TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Vjatseslav Jefimov 176996IAAM

# SMART SERVICE PLANNER APPLICATION DEVELOPMENT FOR INTEST THERMAL SOLUTIONS GMBH

Master´s thesis

Supervisor: Paul Leis

PhD

Tallinn 2019

# Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Vjatseslav Jefimov

12.05.2019

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Vjatseslav Jefimov 176996IAAM

# ETTEVÕTTE INTEST THERMAL SOLUTIONS OÜ NUTIKA HOOLDUSE PLANEERIJA RAKENDUSE VÄLJATÖÖTAMINE

Magistritöö

Juhendaja: Paul Leis

PhD

# Abstract

The aim of this thesis is to find a solution for the problem of routine and inefficient activities in company inTEST Thermal Solutions daily work. Special application, like virtual service planning assistant, could be seen as a solution. The company does not have any information system and a lot of routine operations (including the big data analysis) are performed by the company employees and that is the reason, why service planning is not efficient enought. The company´s revenue should be taken into account, because it highly depends on what marketing strategy decisions were made. The time spent to plan the service, interferes the development of the company strategy. Another aspect is that, if the wrong decision was made during maintenance planning procedure, this can greatly affect the company´s revenue. Summing up, the target is to maximize the amount of time needed for performing the strategic decisions. To eliminate human factor, improve employee performance and increase company revenue, the main maintenance planning tasks of the company should be digitalized, using well-known data scraping and processing methods or modern RPA technology. This thesis outlines the procedure for development a virtual assistant application called „Service Smart Planner“.

The key outcomes of current thesis are the designed IS architecture and application interface prototypes, based on which and using the OpenUP methodology, is possible to start building an information system for the company.

This thesis is written in English and is 43 pages long, including 5 chapters, 19 figures and 4 tables.

# Annotatsioon

# Ettevõtte inTEST Thermal Solutions OÜ nutika hoolduse planeerija rakenduse väljatöötamine

Käesoleva töö eesmägiks on leida lahendus kuidas lihtsustada ettevõtte inTEST Thermal Solutions OÜ hoolduse planeerimist. Käesoleval ettevõttel puudub infosüsteem ning igapäevased toimingud hoolduse planeerimisel teostatakse käsitsi, sealhulgas suure infomahu töötlemisel ja sellepärast ei ole planeerimine effektiivne. Arvestades ka seda, et ettevõtte kasum tugevasti sõltub sellest, mis otsused on tehtud turu- ja müügistrateegia planeerimisel ning aeg, mis on kulutatud hoolduse planeerimise peale, segab müügistrateegia arendamisele. Teine askpekt on see, et hoolduse planeerimise käigus tehtud vea hind on samuti suur. Kui kokku võtta öeldud varem, siis on vaja vabastada rohkem aega strateegia planeerimisele. Selleks, et vähendada inimfaktori mõju, tõsta effektiivsust, tõsta infomahu töötlemise kiirust ja suurendada käivet on vaja enamus hoolduse planeerimise toimingutest digitaliseerida kas kasutades laialtleevinud suure andmetemahu ekstraktimise ja käitlemise meetodeid või populaarsust koguva RPA tehnoloogiat.

Käesolevas töös on välja toodud uue lahenduse metodoloogia valik ja infosüsteemi arhitektuuri arenduseprotsess. Arenduse protsessi käigus oli teostatud sellised toiminguid nagu nõuete välja selgitamine ja kasutuslugude kogumine intervjuu kaudu tueviku kasutajate käest. Välja selgitatud põhirollid ning uue süsteemi aktorid.

Töö olulisemaks tulemusteks on uue infosüsteemi välja töötatud arhitektuur ning kasutajaliidese prototüüp, mille põhjal kasutades OpenUP metodoloogiat hakkatakse infosüsteemi ehitama. Antud töö uue infosüsteemi arhitektuuri arendamise tulemused on kasutusmallide mudel, kasutusmallide kirjeldused, komponentdiagram, evitusdiagram, BPM diagram, järgnevusdiagram ja olemi-suhte diagram.

Uue arenduse teostamise käigus tähelepanu oli pööratud olemasolevate lahenduste taaskasutamisele (näiteks Selenium, AutoIt) ning vana andmete ümberpaigutamise lihtsustamise peale

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 43 leheküljel, 5 peatükki, 19 joonist, 4 tabelit.

# List of abbreviations and terms

| | |
|---|---|
| EU | European Union |
| BPMN | Bussines process modelling notation |
| BPM | Bussines process model |
| UML | Unified Modeling Language |
| SSP | Smart Service Planner |
| https | Hypertext Transfer Protocol Secure |
| IS | Information system |
| SAFe | Scaled Agile Framework |
| RUP | Rational Unified Process |
| IBM | International Business Machines |
| GUI | Graphical User Interface |
| GDPR | General Data Protection Regulation |
| US_ID | User Story identification number |
| BR | Business rule |
| FR | Functional requirement |
| NFR | Non-functional requirement |
| UC_ID | Use case identification number |
| RPA | Ropotic Process Automation |
| OpenUP | Open Unified Process |
| HMI | Human Machine Interface |
| SQL | Structured Query Language |
| XML | Extensible Markup Language |
| API | Application Programming Interface |
| LAN | Local Area Network |
| PC | Personal computer |

# Table of contents

# List of figures

# List of tables

# Introduction

The production of sensitive microelectronics, especially for military purposes, requires from the manufacturer to provide the customer with the quality and reliable product. Unusual and extreme circuimstances in which these kinds of electronics and materials are used to operate, should be created artifically in laboratory conditions. Special testing equipment required to simulate unusual and extreme circuimstances. The company inTEST Thermal Solutions GmbH is producing that special kind of equipment, which allows customers to test their electronics and materials before production. The mentioned equipment (further to simplify the context the „machines“ expression would be used) generates extremely high and extremely low temperatures and provides the customer with a possibility to reduce the transition time from the highest temperature to the lowest within a short time. These machines can generate conditions for electronics, which are typical for military missile. The missile launched first with high speed and high temperature, then it reaches extremely low temperatures in the atmosphere and after (before missile hit the target) it has a high temperature. Another example can be given from civil area like navigation and communication systems in Permafrost regions, where the meteology station workers and scientist are in high dependence of above mentioned systems. The reliability of electronics mentioned above, should be very high. To be able to establish and hold conditions, selected by the manufacturer during the test without any disturbances, machines should have the „under service time“ as short as possible. A lot of companies around European Union (a significant part of them belongs to military field) operates with our machines. Machines are working usually in continuous mode and it is obvious, that all machines, being under operation, require service and if necessary - repairment. For those reasons the support center was established in EU.

The team of EU support center is small and the workload on staff is high because of the limited resources. After becoming a part of the inTEST Thermal Solutions GmbH EU support center team, the author of current master´s thesis noticed, that some operations could be automated and this digitalization/automation can save highly valuable time and costs for service. The author of this master`s thesis will show the way, how can be improved the service procedures, which are focused on customer satisfaction, cost efficiency and moderate workload for service engineers. The aim of the master´s thesis is to prepare the basis for a new application development, which is called Smart Service Planner.

The relevance of current master`s thesis topic is high, because the european service center revenue depends on the amount of installed machines and satisfied customers. It means that, if there is a possibility to cover as much customers as possible and perform a service and repairment jobs as fast as possible, then we can attract much more new customers and sign a contracts for more machines with current customers.

During the analysis and development of mentioned application the common methodologies, BPMN and UML language were used. For the BPM visualisation and improvement the „Bizagi“ software was used.

In the first part of current master`s thesis author will describe the relevance of current master´s thesis topic more precisely and will show disadvantages for the company because of the missing information system.

In the second part of current master´s thesis author will give the aim about the methodologies and techniques used during SSP development. To perform successfully development of a new solution, the most suitable methodology will be chosen. This part will describe the suitable methodology selection for SSP development.

The third part is the main part. After starting of IS analysing, it was obvious, that IS in its usual sense, was missing, only server with sorted files. Storage/stock system was presented as a separate application. Third part has a similar structure as a project, followed the OpenUP methodology rules and prescriptions. The author of current master`s thesis will develop SSP, following OpenUp methodology rules. The third part will contain a service smart planner architecture view, deployment diagram, component diagram, sequence diagram, entity-relationship diagram and also description of application interface prototype. In other words author will create diagrams and documents, which are required for successful IT-project starting. Additional, the author will try to show the example of regular iteration procedure.

# 1 Description of the problem

Working in situation, when circuimstances are changing very rapidly and accidentally, increases a human factor appearance probability. Concerning current company structure, there is one person, who is responsible for the resources and service visits planning. Taking into account, that number of customers around the European Union (also some of them outside of EU) is large and they are spreaded over the large area, it happens quite often, than the service is planned not in efficient way (additional cost for traveling and lost time added). There should be mentioned also that for the customers is common to change their mind at the last minute after service visit arranged. Another aspect is that there are several engineers, with different experience and skills, who can do different kind of service jobs with different accuracy and speed. This should be also taken into consideration, during resources relocation between customers, due to changes in their plans. Last, but not least aspect is a need for special parts nad special tools for different kind of jobs. Of course, if engineer is already experienced and knows his job good, he do not need to spend much time for planned trip preparation, but even experienced people can forget something sometimes (human factor). As there was mentioned above, the location of the customer might be far away from service center and stock, and if there is missing something important for the service job, there could be a huge loss of time and money. This is the reason, why human factor needs to be lowered as low as possible or eliminated totally. Also should be not forgotten the Force Majeure, like delay in flights or traffic jams on highways. All aspects mentioned above draw a clear picture, that current company service planning procedures are changing in real time, they are complex and have a high price for possible mistake or inefficient planning. By inefficient planning I mean, for example, following situation: the service job should be performed in two locations, which are relatively close to each other, but service jobs are organized in way, that services are covered separately from each other, like two round flights (4 flights, to the customer and return flight) instead of one 1 round flight. Of course it is very simplified example and there could be more that 5 customers during one week, which makes this kind of planning very complex. The only one possibility to totally eliminate the human factor is to delegate to the application (information system) some operations, or in another words to digitalize the service manager and service engineer routine and complex operations.

## 1.1 The goal of current thesis

The main goal of current master´s thesis is to think out, develop and prepare information architectural basis for new information system development, show examples of project iterations and present user friendly prototype interfaces.

## 1.2 The relevance of the problem

The relevance of current solution for the company is considered from the prespective, that the hiring additional staff in the company is not reasonable due to limited financial resource and divided tasks between current team members. But the load to centre director and further to service manager is high. The revenue of the company consist of custumer service incoming payment and sold machines income. As already was mentioned, the cost for travel can be highly reduced and more free time could be released for center director and service manager. This released time may be used to promote product to new customers and to strengthen the market relationships between current customers and the company.

The automated routine operations and procedures can eliminate the human factor from the service planning, so released free time for centre director and service manager gives a possibility to concentrate on strategic desicions. It means that the amount of sold mashines could be higher due to better strategic desicions and service plan could have a service jobs performed with higher accuracy and there would be covered more customers than usually.

## 1.3 Chapter Summary

The routine daily operations with huge amount of continuously changing data in corresponding company is not efficient, if these operations are performed manually by current company limited labor resources. High valuable time, spent to plan the service jobs by the company service manager, does not give him a chance to concentrate on strategic marketing desicions. Also time, spent by the company engineer to prepare tools and parts, does not give him an opportunity to concentrate on technical aspects of the customer problem.

# 2 Methodology selection

To start, perform and finalize the project successfully, the right methodology should be selected before project starts. There were analized several aspects of the corresponding company. The first aspect concerns labor and time resources. At the beginning of discussion of the current topic, there were two options how to realize the project. First one was to perform all stages of project by limited local resources. It means, that implementation, testing and release of the current new developed product would be carried out by author of current master thesis. It requires on the one hand the less material resources, but much more time resources and on the other hand high risks might appear. Another option was to use an outsourcing companies to carry out current project. It requires more costs for workpower, but the level of risks and time period would be much more lower. Finally, after discussion, there was decided not to force the performance of implementation, testing and release stages and give more time to develop new solution with reasonable limited team (SAFe frameworks can not be used). It means, that current development project would be in time aspect long project, but at the same time it would be with high formalization and with precisely defined business rules and requirements. Taking into account, that current solution has a lot of exceptions, there was decided to implement each of several exceptions as a new feature to the project. That means, that selected methodology must be iterative (no „Waterfall“ methodologies can be selected). Also „Scrum“ methodology can not be selected, because it focuses on short and middle term projects. The best value can be delivered by unified process models like RUP or OpenUP. RUP is to „heavy“ for team with limited resources and belongs to IBM. OpenUp is the special case of RUP and free to use with simplified lifecycle. In current solution the OpenUp methodology was chosen. The main part of current thesis contains the main OpenUP lifecycle stages. Each iteration consist of four stages: Inception, Elaboration, Construction and Transition. The transition stage was not included in current work.

## 2.1 Chapter Summary

A lot of exceptions in corresponding solution will exclude the „Waterfall“ variant from the scope. In the same time the duration of current development will be long because of the limited labor resources, and that will exclude some iterative methodologies like „Scrum“. Current development has a high formalization level and precisely defined business rules and requirements at the initial phase of the project. The RUP methodology fullfills all requirements for current development, but it is to „heavy“ for small companies and not open-sourced. The alternative and the simplified form of RUP - the OpenUP open-sourced methodology was chosen for current work development.

# 3 SSP development

New application development will be started with description of the system preconditions and will continue to follow the selected OpenUP methodology phases. The initial state of business flow will be represented graphically on BPM diagram in *Initial state description* chapter.

## 3.1 Development preconditions

To be able to make a changes in business action flow, the graphical representation is required. *The Bizagi Modeler* was used to draw BPM diagram.

### 3.1.1 Initial state description

Before development of a new solution begins, there should be determined the initial state of the current company structure and business flow. To show the current business flow the UML language was used. On the next picture is presented the company main activityflow:

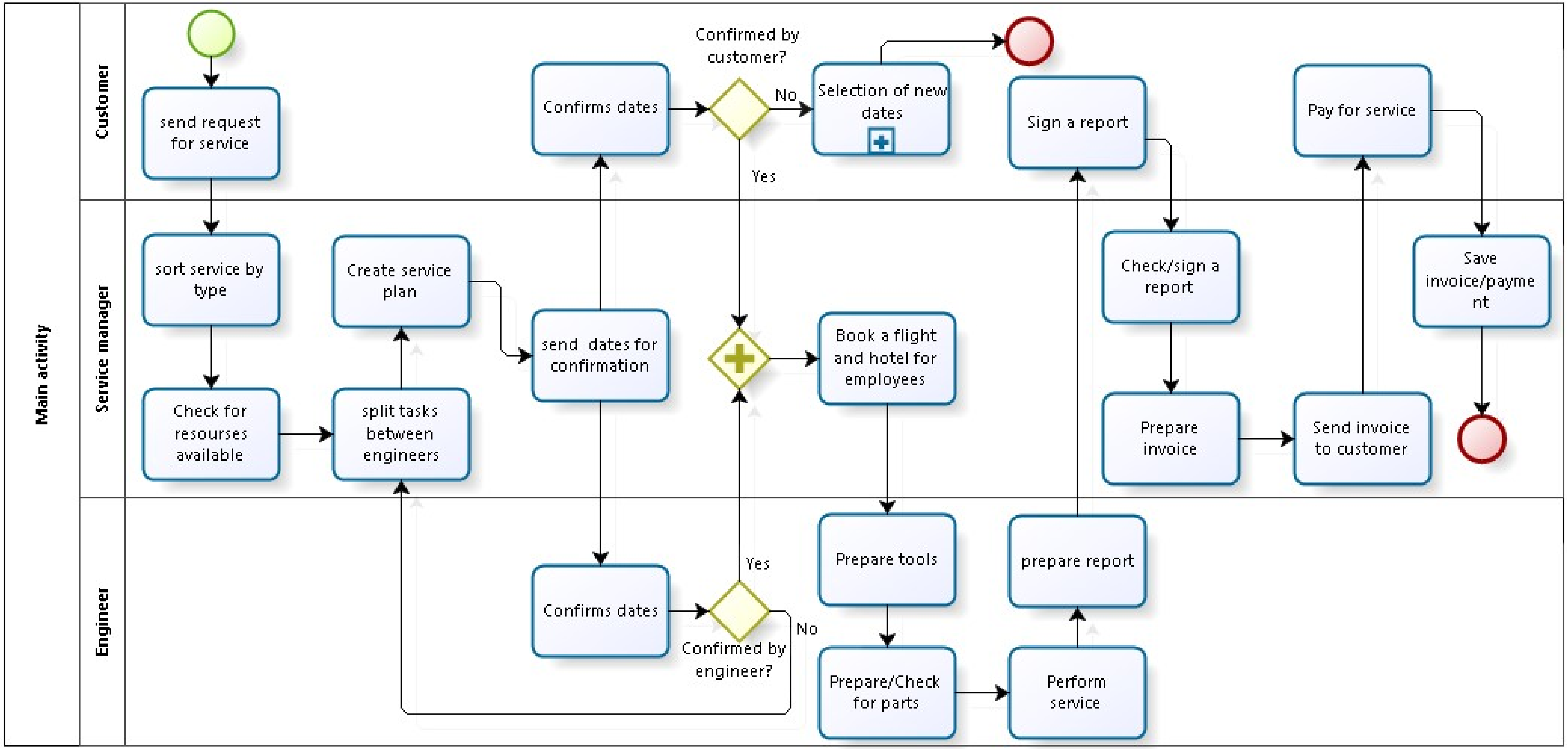


**Figure 1.** Main activity BPM.

Presented diagram is too detailed, because author tries to show the whole organisation main service planning steps and all involved actors. The diagram shows that, due to absence of appropriate IS a lot of actions should be performed by service manager. This makes overall business inefficient. To eliminate inefficiency there is a need to delegate the rights to maintain major activities to new developed IS. The critical aspects are additional time resource and human mistake probability in that kind of structure. Before starting to follow a lifecycle of IS development I need to determine and give a name to the future application. As I am going to improve service planning procedures in corresponding company and digitalize main activities the name of future IS would be „Smart Service Planner“. Further there would be used abbreviation „SSP“.

## 3.2 New solution development key components

As described in previous chapters the focus of current development is to create a virtual service assistant. In functionality of virtual service assistant should be included following statements:

1) The virtual assistant should be able to read needed information from predetermined sites. This can be realized by using RPA techology, data scraping and parcing method (Examples: Selenium, OutWit Hub).

2) The virtual assistant should be able to do typical user actions on web pages like inserting/typing some predefined text in proper field, click on proper buttons and so on. This can be realized by widely used GUI scripting language automation software (Examples: „Autolt“)

3) The virtual assistant should be able to represent collected, sorted and structured information to end user (service manager, service engineer)

4) The virtual assistant must be controlled in easy and simple way, and operate in online mode (to be able to change plans dynamically).

Should be mentioned also GDPR directives. Due to the fact that data, which is in focus of our IS data scraping engine is not a personal or sensitive, there could be no restrictions from the law to use and collect needed data for service. There can be only restriction to use data scraper or web crawler, because it can make slower the webservice performance.

SSP development should follow OpenUP rules. OpenUP applies the Unified Process phases: Inception, Elaboration, Construction, and Transition. All these phases presented in each methodology iteration. The most detailed and deep phase should be first, because there is a requirement to keep a formalization level high and determine precisely all business requirements, architecture, key system functionality iteration plan. Before development starts, there were performed interviews with potential users of future IS. Based on answers and expectations of end users the following list of user stories was created:

**Table 1**. User stories.

| Use story ID | As a/an | I want to... | So that.. |
|---|---|---|---|
| US_ID_1 | Service manager | Check available employees | I can plan a service jobs |
| US_ID_2 | Service manager | Check available parts on stock | I can plan a service jobs |
| US_ID_3 | Service manager | Know available hotels close to customer | I can plan a service jobs |
| US_ID_4 | Service manager | Know available flights | I can plan a service job |
| US_ID_5 | Service manager | Be informed about service time, prefered by the customer | I can plan a service job |
| US_ID_6 | Service manager | Be informed about changes in customer plans | I can plan a service job |
| US_ID_7 | Service engineer | Prepare needed tools | I can do a service job |
| US_ID_8 | Service engineer | Have access to stock system | I can remove/add/order components |
| US_ID_9 | Service engineer | Have access to document centre | I can follow the service history |
| US_ID_10 | Service engineer | Have administration rights | To change documents on server |

## 3.3 Inception

In the Inception phase there are several objectives like understand what to build, identify key system functionality and determine at least one possible solution.

### 3.3.1 Define business rules

Business rules were partially created in chapter “New solution development key components“. In this chapter the business rules were composed in more precise way.

BR_1: The virtual assistant should be able to read needed information from predetermined sites.

BR_2: The virtual assistant should be able to do typical user actions on web pages like inserting/typing some predefined text in proper field, click on proper buttons and so on

BR_3: The virtual assistant should be able to represent collected, sorted and structured information to end user

BR_4: The virtual assistant must be controlled in easy and simple manner, and operate in online mode

BR_5: The customer can change preselected service dates not later then one week before service plans were established.

The rules above are initial rules, on next Inception phase iterations business rules would be updated.

### 3.3.2 Define business roles and actors

The significant part of the Inception phase is to define balance competing priorities to maximize stakeholder value.

In OpenUP methodology there are following roles presented: Stakeholder, Analyst, Architect, Developer, Tester, Project Manager and Any Role. Taking into account, that corresponding company does not have enough employees to cover all roles, which are determined in OpenUp methodology, the participants of this project have several roles. The mentioned roles divided between company´s employees involved in current master´s thesis are listed below:

- Author of current master´s thesis (position in the company: service engineer) is represented in following roles from list above: Analyst, Architect and Any Role.
- The current company general service manager is represented in following roles from the list above: Stakeholder, Project Manager and Any Role.
- The outsourcing developer has the following roles: Developer and Any Role.

Define actors:

In new solution there would be four main Actors:

- Customer
- Service Manager
- Service Engineer
- SSP

The use-case diagram with all listed actors in previous statement, would be presented in phase Elaboration.

### 3.3.3 Define functional and non-functional requirements

Based on defined business roles and key components the functional requirements of the SSP were listed below:

**Functional requirements:**

**FR_1:** Recovery ability of IS: The IS should be able to save all configured data after accidental switch off the system.

**FR_2:** Controll of the IS: The IS should always ask for confirmation after user click on button “confirm”.

**FR_3:** Backup of the IS: The IS should save at least 30 last configured service plans.

**FR_4:** Action assist of IS: The IS should deactivate already selected and confirmed dates, under manual reconfiguration of the service plan by the user

**FR_5:** If the user/service manager is not agree with the preconfigured plan, he can change the configuration settings (for example change time/price priority) before next configuration.

**FR_6:** All user and IS actions should be saved in log file.

**Non-functional requirements:**

**NFR_1:** Speed of IS: The time for reconfiguration after service dates were changed, should be not more then 30 seconds.

**NFR_2:** Reconfiguration of IS: The IS should start to reconfigure the service plan immediately after customer changes service date.

**NFR_3:** Reporting of IS: The IS should inform service manager about reconfiguration immediately after customer changes service dates.

**NFR_4:** Representation of IS: The system should represent the collected sorted and structured information

**NFR_5:** Recovery of IS: The IS should be able to restart itself after system crash.

**NFR_6:** IS log file should have a possibility to store at least 10 MB of information about user and IS activity.

All functional and non-functional requirements would be updated during next OpenUp Inception phase iterations.

### 3.3.4 Define possible solution

To determine and describe a possible solution, the BPM diagram was created, using *Bizagi modeller* software. The BPMN notation was used. During the creation of BPM diagram the author tried to make last more simple. Next picture is representing possible solution of the SSP IS.

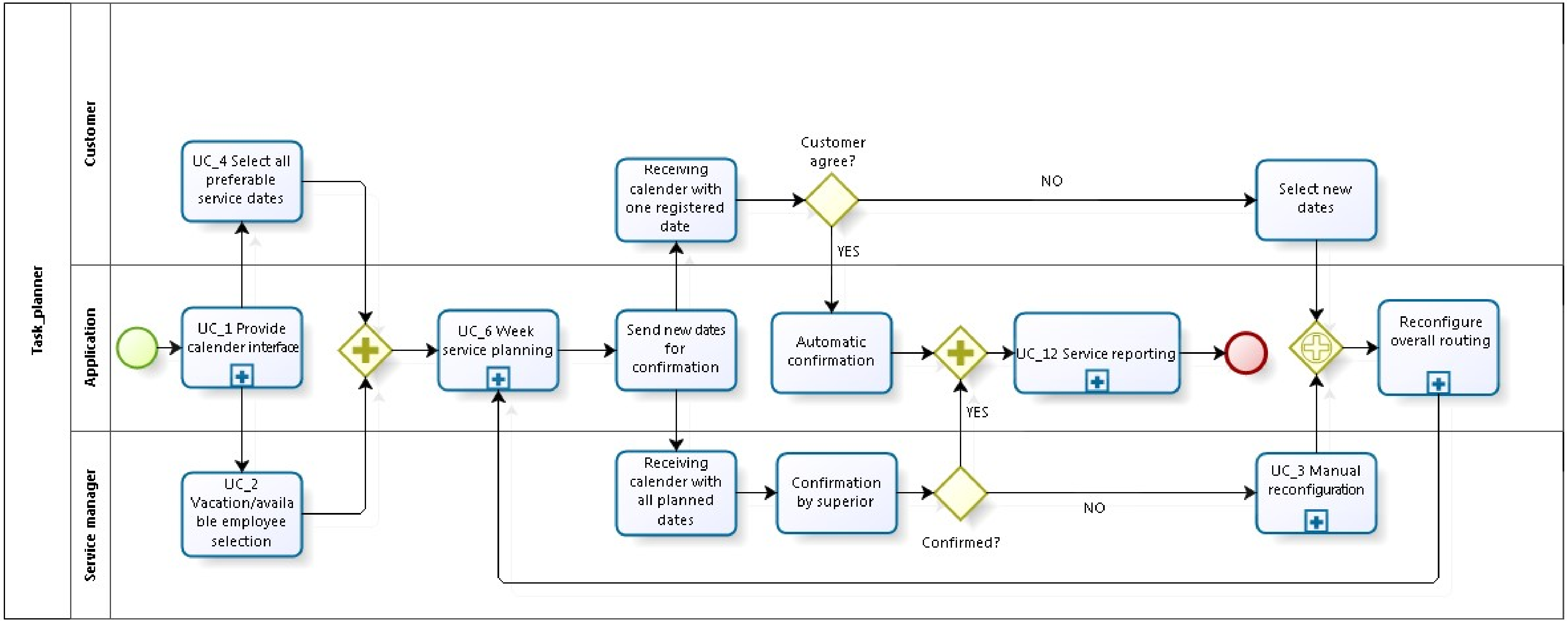

Task_planner
Customer
Application
Service manager
UC_4 Select all preferable service dates
Receiving calender with one registered date
Customer agree?
NO
YES
Select new dates
UC_1 Provide calender interface
UC_6 Week service planning
Send new dates for confirmation
Automatic confirmation
UC_12 Service reporting
Reconfigure overall routing
YES
UC_2 Vacation/available employee selection
Receiving calender with all planned dates
Confirmation by superior
Confirmed?
NO
UC_3 Manual reconfiguration

**Figure 2.** Task planner BPM.

### 3.3.5 Determine the scope

As there was no information system presented in corresponding company, except the stock system/interface and stored documents in usual document-file form on company server, the new solution should be developed from the beginning and the scope and risks of future work must be determined. Below are listed main project aspects:

- Costs

  Development of SSP based on current master´s thesis is planned to be performed by the author and small outsourcing team, consist of maximum two developers. The stakeholder, as was mentioned in previous chapters, is company director. Taking also into account, that there is a plan to use existing solutions (freeware, open source code) to develop our SSP virtual assistant, the pay-back period after implementing solution, described in current master´s thesis, would be not longer then one year.

  Risks: The code/logic writing may be more difficult as it seems at the begining. In this case the development team should be highly professional.

- Time

  The most critical in current development is time, because of the limited resources and lack of knowledge and experience in current field. Usually, that kind of projects lasts not longer then 5-6 months, but in our case the development period should be extended to one year.

  Risks: The development procedure may be longer as expected.

- Resouces

  Resources are limited and as was mentioned before, the project team will consist of 4 people.

  Risks: Developers, who will participate in current project, are most experienced in IT field and together with longer project period there is a possibility that some of them refuse to participate in one project so long.

The potential cost savings after implementing ready solution were approximately calculated. The positive impact of future solution may be high enough, taking into account than some spare parts may cost for the client several thousands euros and travel cost in one week may be even higher.

## 3.4 Elaboration

In phase Elaboration the more detailed understanding of the requirements is presented, realized the design of the architecture and use case diagram. During first iteration the use-case description performed, the use case diagram created, presented initial defined architecture of the new solution, BPM model for the key use case, component diagram and deployment diagram. Next were created sequence and ERD diagrams. During first iteration of Elaboration phase the first version of prototype was created. The Elaboration phase contains the major effort to design the architecture of a new solution.

### 3.4.1 Define use-case

Based on collected user-stories the use-case description created. First there was described the main business flow of a new solution below.

**Table 2.** Task planner flow description.

| | |
|---|---|
| Use case name | Task planner |
| ID | UC_ID_00 |
| Description | The application provides calender interface for the client and for the company service manager, where client company representative can select and confirm dates for service |
| Actors | Customer, Application, Service manager |
| Triggers | |
| Preconditions | Client and company representative allowed/installed current application |
| Postconditions | Client and company can follow and plan future service jobs |
| Main Course | 1.Client opens an application.<br>2.Client select service dates preferable for him<br>3.Company representative opens an application.<br>4.Company representative select dates with available employees.<br>5.Application configures the bussiness trip and sends to the customer and to the company representative the new calender with reconfigured dates |

| | |
|---|---|
| | 6. Customer confirmes a dates which were first selected by him and then the best suitable of them were chosen by the application<br>7. Service manager confirms selected dates.<br>8. Application generates the notification for employees |
| Alternate Courses | 1.Client opens an application.<br>2.Client select service dates preferable for him<br>3.Company representative opens an application.<br>4.Company representative select dates with available employees.<br>5.Application configures the bussiness trip and sends to customer and to the company representative the new calender with reconfigured dates<br>6. Customer change the decision and select dates for the next service period.<br>7. Service manager decline proposed dates and manually add other service dates.<br>8. Application reconfigures overall routing and return to the UC_ID_00 |
| Exceptions | To be considered in next iterations |

In current master´s thesis not all use-cases would be described, only those, which have a significant value to explain the operation principle of SSP. The second and the most important use case, which contains the main logic of SSP operation is presented below.

**Table 3.** Use case “Week service planning” description.

| | |
|---|---|
| Use case name | Week service planning |
| ID | UC_ID_06 |
| Description | Application establish the less expensive, shortest possible route and includes as much service jobs as possible in one planned week. |
| Actors | Application, Service manager |
| Triggers | |
| Preconditions | Application installed, all preferable dates by the customer selected, all dates with available employees are selected by service manager |
| Postconditions | Next service week plan generated |
| Main Course | 1. Application receive selected dates<br>2. Application sorts services by location<br>3. Service manager select the priority for the tasks<br>4.Application sorts service dates by priority.<br>5. Service manager select service type<br>6. Application sorts service jobs by type.<br>7. Application inserts data into predefined websites (imitaton of user actions) |

|  |  |
|---|---|
|  | 8. Application check the stock for needed parts.<br>9. Application performas a web crawling and data scraping.<br>10. Application establish shortest route and time for service trip.<br>11. Application organize the closest and cheapest (according to preadjusted price/distance balance) overnight stay.<br>12. Application add needed parts into service list.<br>13. Application sends the generated service dates to service manager. |
| Alternate Courses | 1. Application receive selected dates<br>2. Application sorts services by location<br>3. Service manager select the priority for the tasks<br>4.Application sorts service dates by priority.<br>5. Service manager select service type<br>6. Application sort service jobs by type.<br>7. Application inserts data into predefined websites(imitaton of user actions)<br>8. Application check the stock for needed parts.<br>9. Application can not find needed spare parts.<br>10. Application sends notification for service manager<br>11. Service manager orders needed parts<br>12. If the delivery time more then 7 days the application will remove current task from service plan |
| Exceptions | To be considered |
|  |  |

### 3.4.2 Use-case model

New solution´s use-case model is represented on the next picture. The use-case diagram is based on use-case descriptions and was created in StarUML software.

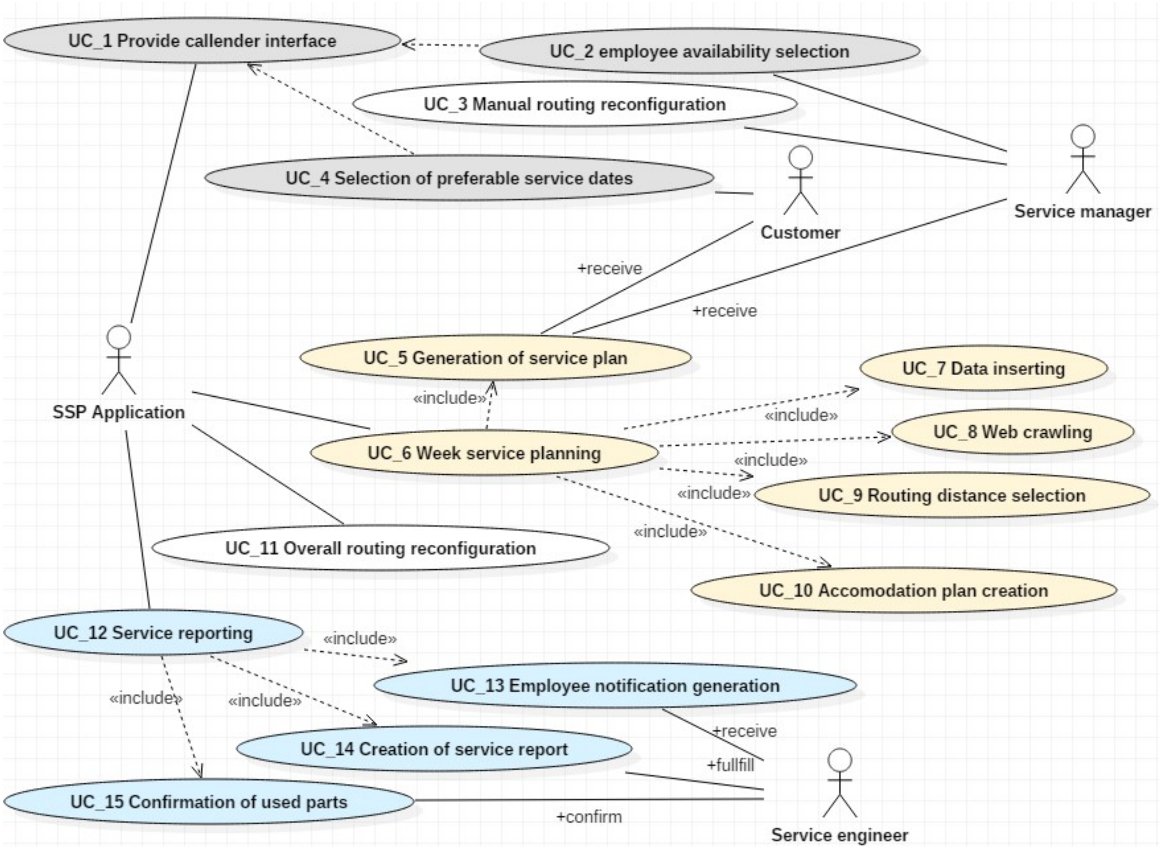

UC_1 Provide callender interface
UC_2 employee availability selection
UC_3 Manual routing reconfiguration
UC_4 Selection of preferable service dates
Customer
Service manager
+receive
+receive
SSP Application
UC_5 Generation of service plan
«include»
UC_6 Week service planning
UC_7 Data inserting
«include»
UC_8 Web crawling
«include»
«include»
UC_9 Routing distance selection
«include»
UC_11 Overall routing reconfiguration
UC_10 Accomodation plan creation
UC_12 Service reporting
«include»
«include»
«include»
UC_13 Employee notification generation
+receive
UC_14 Creation of service report
+fullfill
UC_15 Confirmation of used parts
+confirm
Service engineer

**Figure 3**. Use-case diagram.

The colors used in picture above shows actors relation density to different use cases. The light blue use-cases concerns interconnection between service engineer and SSP application. Yellow use-cases belongs mostly to SSP application.

The initial use-case diagram may be updated during next iterations.

### 3.4.3 Define the Architecture

The focus on the architecture as early as possible to minimize risks and organize development in manner to eliminate possible future errors. Below is presented the most interested use case in current work scope detailed UI_ID_6.

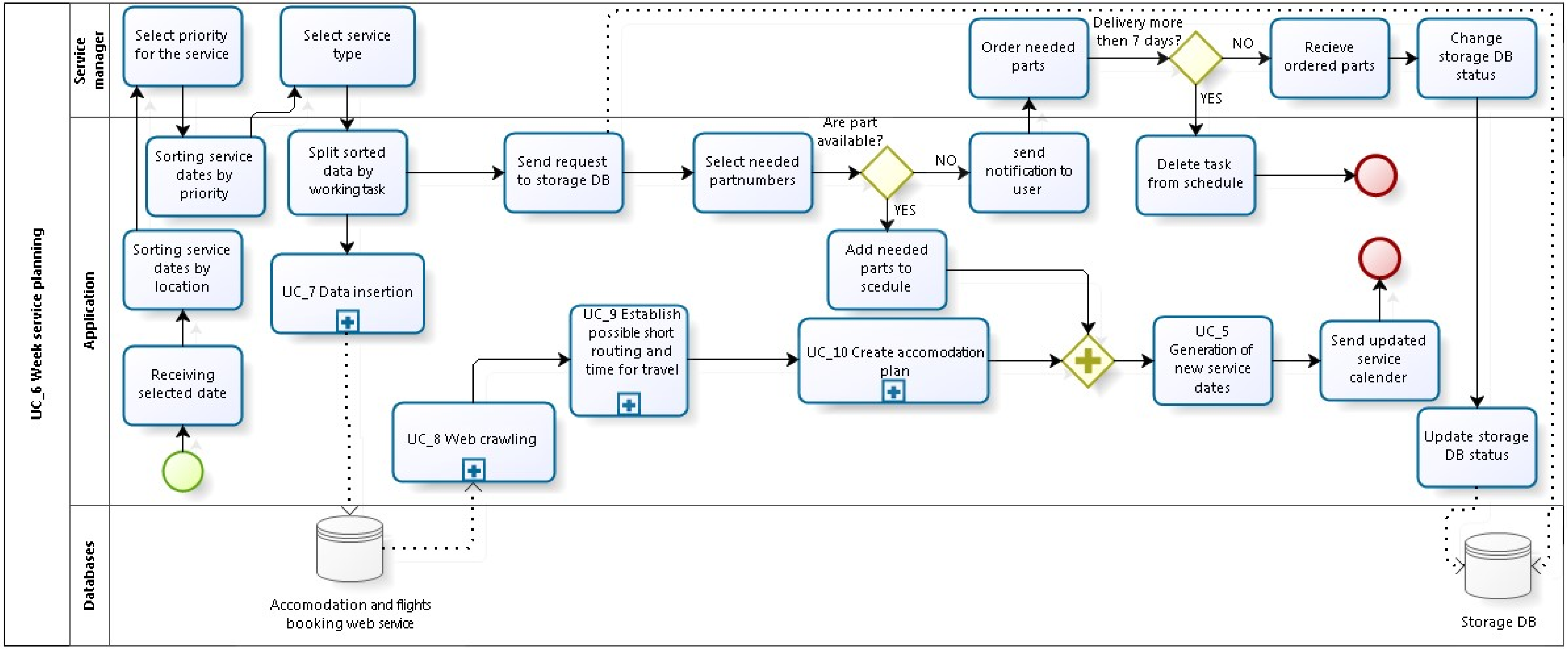

UC_6 Week service planning
Service manager
Application
Databases
Select priority for the service
Select service type
Sorting service dates by priority
Split sorted data by working task
Sorting service dates by location
Receiving selected date
UC_7 Data insertion
Send request to storage DB
Select needed partnumbers
Are part available?
NO
YES
send notification to user
Order needed parts
Delivery more then 7 days?
NO
YES
Recieve ordered parts
Change storage DB status
Delete task from schedule
Add needed parts to scedule
UC_9 Establish possible short routing and time for travel
UC_8 Web crawling
UC_10 Create accomodation plan
UC_5 Generation of new service dates
Send updated service calender
Update storage DB status
Accomodation and flights booking web service
Storage DB

**Figure 4.** Use case "Week service planning" BPM diagram.

One remark should be mentioned, that Web crawling in current context means also data scraping and parcing. In other words, SSP first inserts data (simulates the user action) and then collects the results. Next picture represents the Component diagram of SSP:

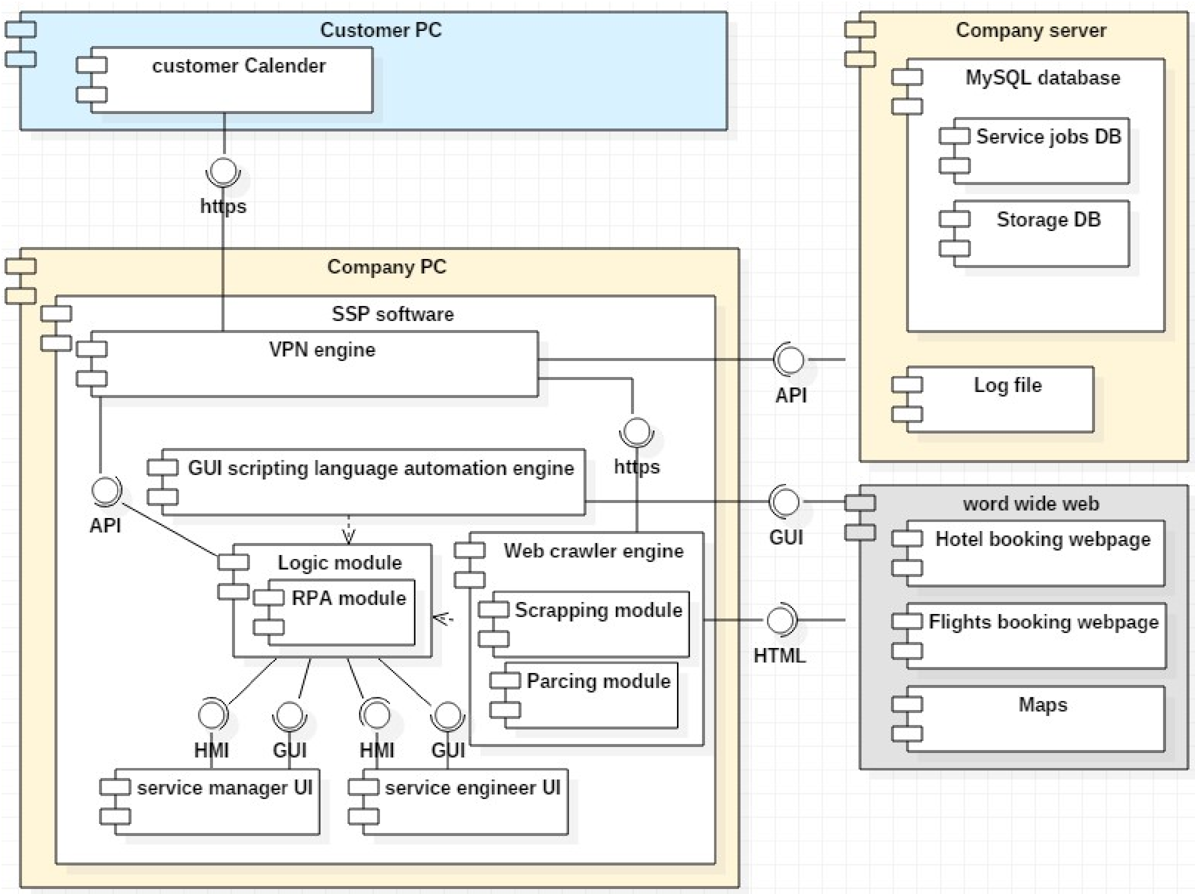


**Figure 5.** SSP Component diagram.

The key components of current component system are logic module with RPA module, GUI scripting language automation engine and web crawler engine. GUI scripting language automation engine includes the code logic for simulating user interaction in websites. For the basis can be selected freeware “AutoIt”. The script of AutoIt can be reusable in our solution with some minor modifications. The next important module is “web crawler engine”. In current context this module includes the logic of web crawling, data scraping and parsing. There are two ways to scrap a data from graphical website interface: to read the HTML and extract needed text from there or use the screen scraping technique. In current solution the second way is preferable. In our case we use both methods, even if screen scraping is preferable. The information cathered by screen scraping, are compared with information, extracted from parced HTML and XML. Some simple actions can be performed first by the user and using RPA technology will train the software robot to perform them automatically in the future. Logic module makes possible for GUI sctripting language automation engine and Web crawling module to operate together. As was described in the beginning of current master´s thesis, the SSP first inserts required/predefined information by the customer and service manager/director into websites and after that browses and copy needed information for the service plan creation. To insert predefined information and to receive ready service plan details GUI interfaces are used. There are two different SSP interfaces for service manager and service engineer. Taking into account that on customer side there is no possible to install SSP, the customer provides the service team with snapshot of calender with selected service dates. To follow the safety requirements, data is collected via virtual private network from client and from company servers.

The next picture represents the deployment diagram of new information system.

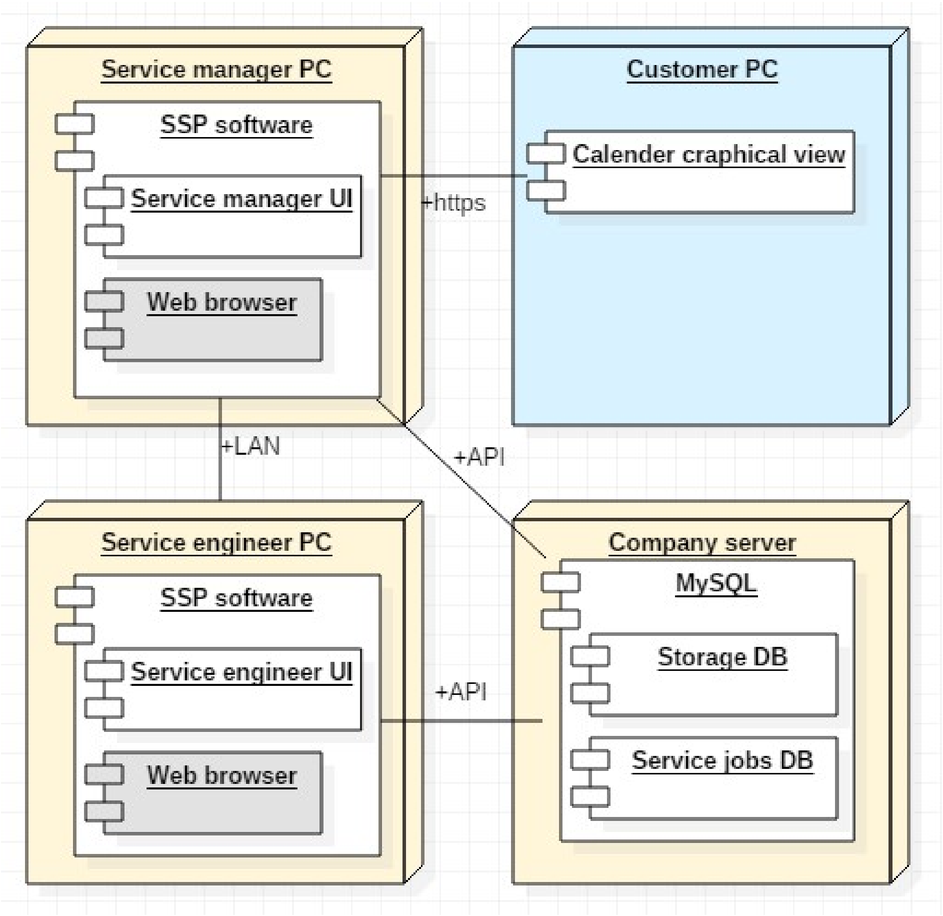


**Figure 6.** SSP Deployment diagram.

The used color codes represents the connecton between component diagram and deployment diagram. There are three color codes, which emphasis the belongings of components (component diagram) and nodes (deployment diagram) to the company, client or word wide web. There are four main nodes: service manager PC, service engineer PC, company server and customer PC. The SSP installed on service manager and service engineer PC. The customer PC has only graphical representation of calender with suitable service dates. The company server is connected to service manager and service engineer PC via application programming interface.

Next step of architecture development is to show the sequence of messages, exchanged between the main objects and it time sequence. The SSP sequence diagram was created in StarUML and is represented below.

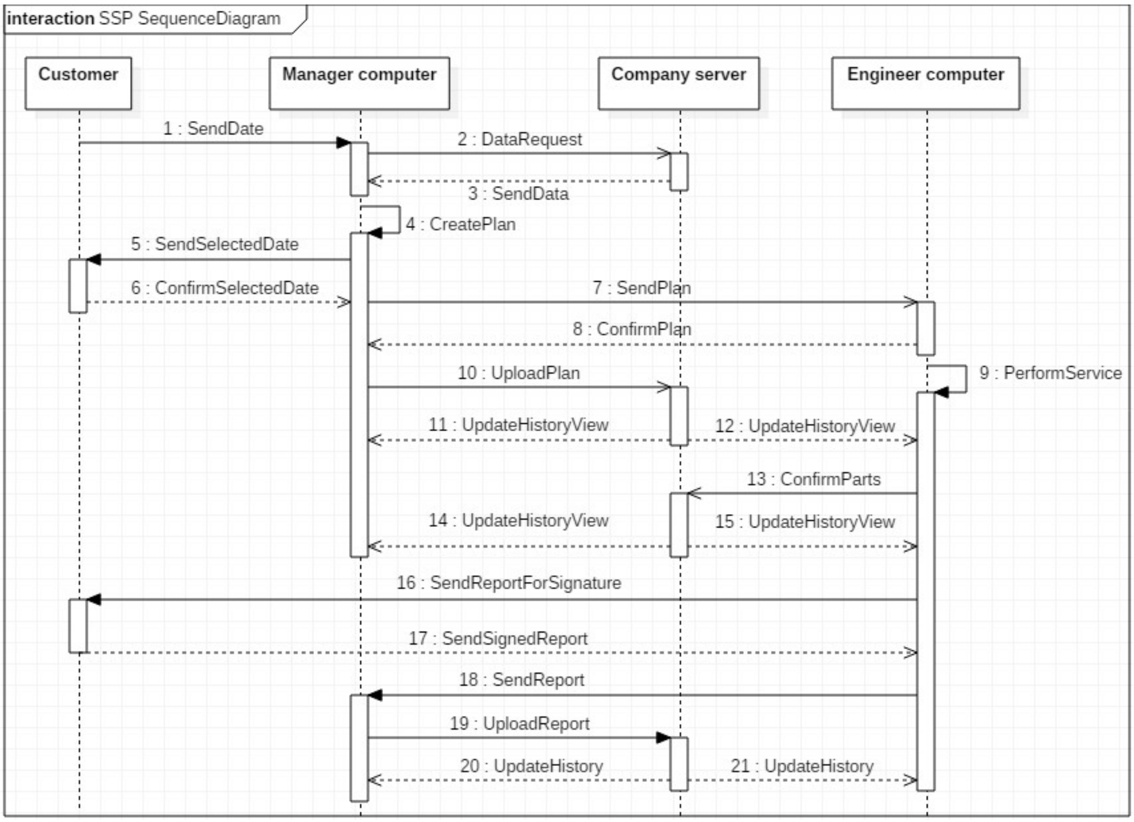


**Figure 7.** SSP Sequence diagram.

The interaction starts, when the client send the preferred dates of service to SSP which is installed on manager computer. SSP requests data from the company server. After receiving required data, SSP starts to develop new service plan and, when finished, sends the selected service date to the customer. After confirmation of service date, the SSP activates the service plan on engineer computer. Engineer confirms the service plan. SSP uploads service plan to the company server and then a new record is saved in SSP interfaces. When service completed, engineer receive the signature from customer and report will be sent to manager SSP. Manager SSP uploads it to company server.

After each of interactions mentioned above the server sends a command to SSP to refresh information indicated on SSP interface.

The data architecture is one of the most important part of current development. To define the information structure, the entity-relationship diagram was built, using the QSEE SuperLite software. On the next picture is represented complete SSP ERD model.

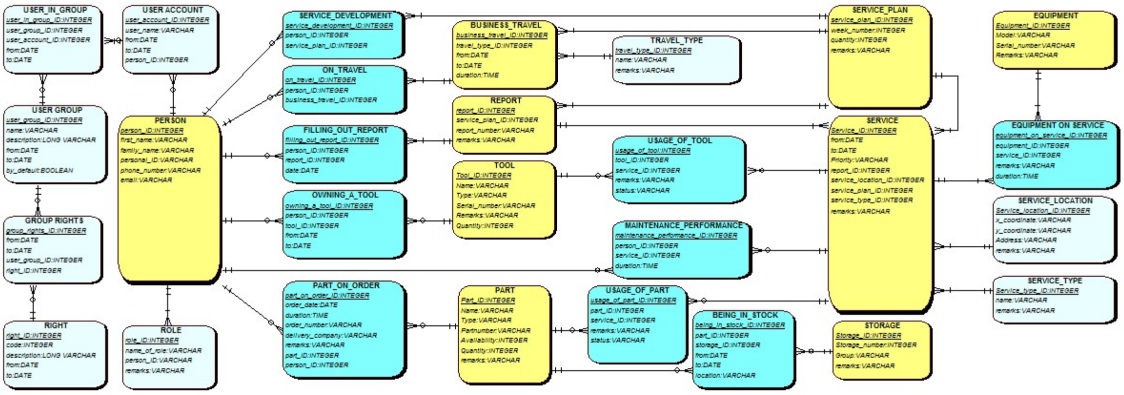


**Figure 8.** SSP ERD complete**.**

The created ERD model is complex and large and to make it readable the model showed above was split up into two parts (Figure 9 and Figure 10). Description of color codes used in ERD model:

- main entities- yellow
- main interconnection entities- neon blue
- other entities- grey

The semantics of SSP ERD model is represented in table below (Table 4).

**Table 4.** SSP ERD Semantics.

| Entity | Description |
|---|---|
| Person | Information about user |
| Role | Information about user role (manager, engineer) |
| User account | Information about user account in SSP |
| User in group | Information about temporarily belonging of the user to the user group |
| User group | Information about user group (manager on engineer user group) |

| | |
|---|---|
| Group rights | Information about user group rights (set of manager rights, set of engineer rights) |
| Right | Information about rights (change service plan, select service engineer, select priority to the service) |
| Service plan | Information about service plan (service plan week number, quantity of service jobs) |
| Service development | Information about service planning |
| Business travel | Information about service travel (when, how long) |
| On travel | Information about determined engineer in service travel |
| Report | Information about service report |
| Filling out report | Information about completed reports |
| Tool | Information about tools, used in service (type, serial number, quantity) |
| Usage of tool | Information about being tool in service (who use the tool and how long) |
| Owning a tool | Information about belonging of the tool to determined person |
| Service | Information about service (priority of the service and execution time) |
| Maintenance performance | Information about who from engineers will perform a service and how long |
| Part | Information about parts (name, type, partnumber, availability, quantity) |
| Part on order | Information about ordered part (order date, duration of delivery, order number, delivery company) |
| Usage of part | Information about part being used in service |
| Storage | Information about company storage (storage number and storage group) |
| Being in stock | Information about part location on the company stock (location and storage time) |
| Equipment | Information about company´s sold products (model and serial number) |
| Equipment on service | Information about equipment under service |
| Service location | Information about locaton of customers equipment under service (address, coordinates) |
| Service type | Information about service type(installation, repair, callibration) |
| Travel type | Information about service travel type (by plane, rental car, company car) |

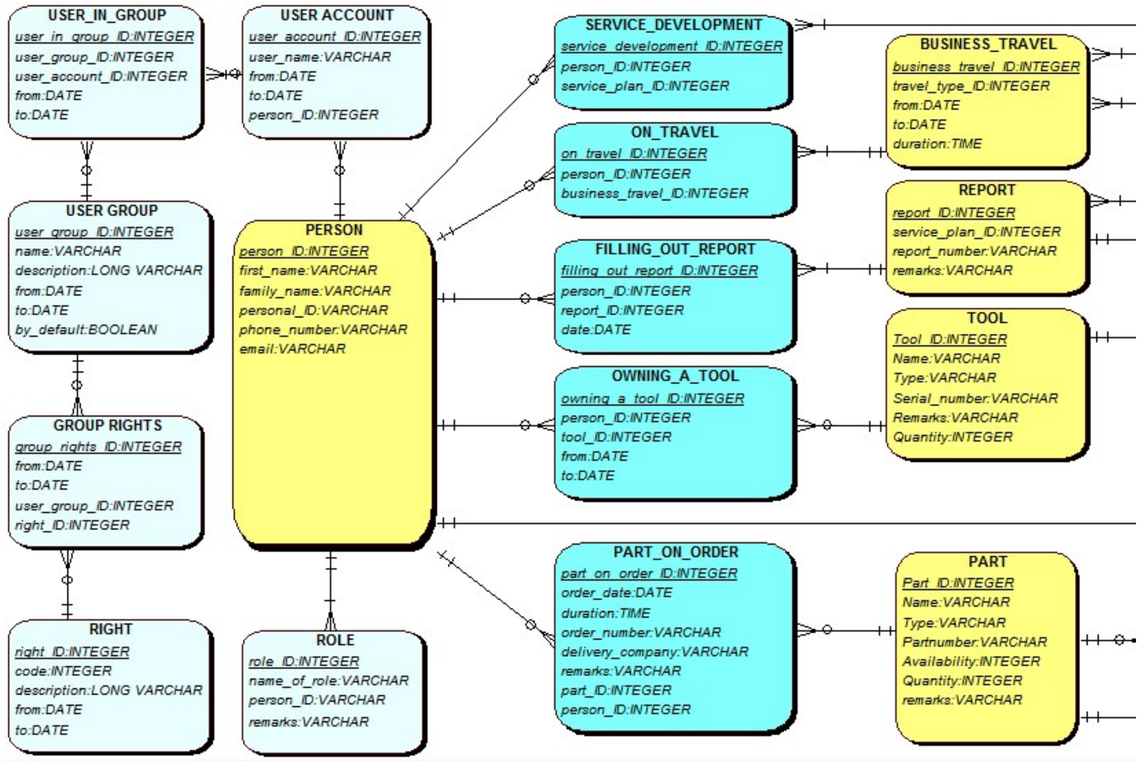


**Figure 9.** SSP ERD first part.

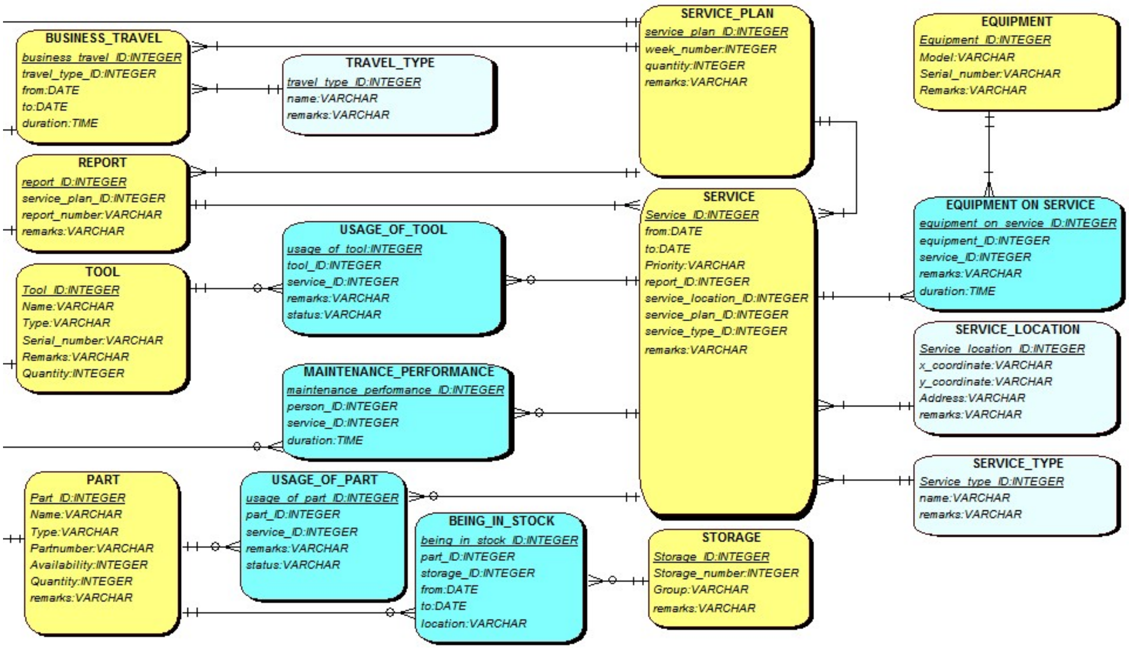
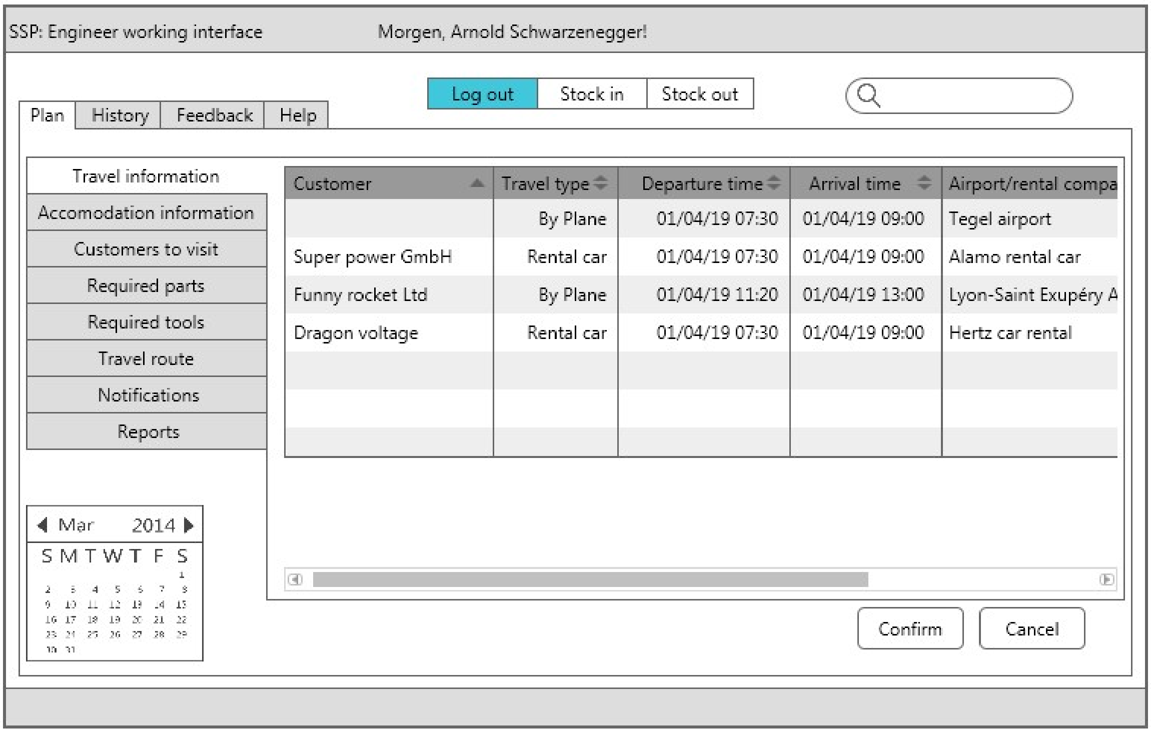


**Figure 10.** SSP ERD second part.

### 3.4.4 Prototyping

The purpose of prototyping is to create possible vision of future operational interface for current company employees. The prototypes were created using Mockplus software. The main goal is to develop a user friendly and at the same time multifunctional interface. According to divided roles, there should be presented prototype with two interfaces: engineer and manager interfaces. There were decided to present first the engineer SSP interface prototype (which is more simple then manager SSP interface prototype) and then manager SSP interface prototype to show the main differences. On the next picture presented engineer interface prototype.



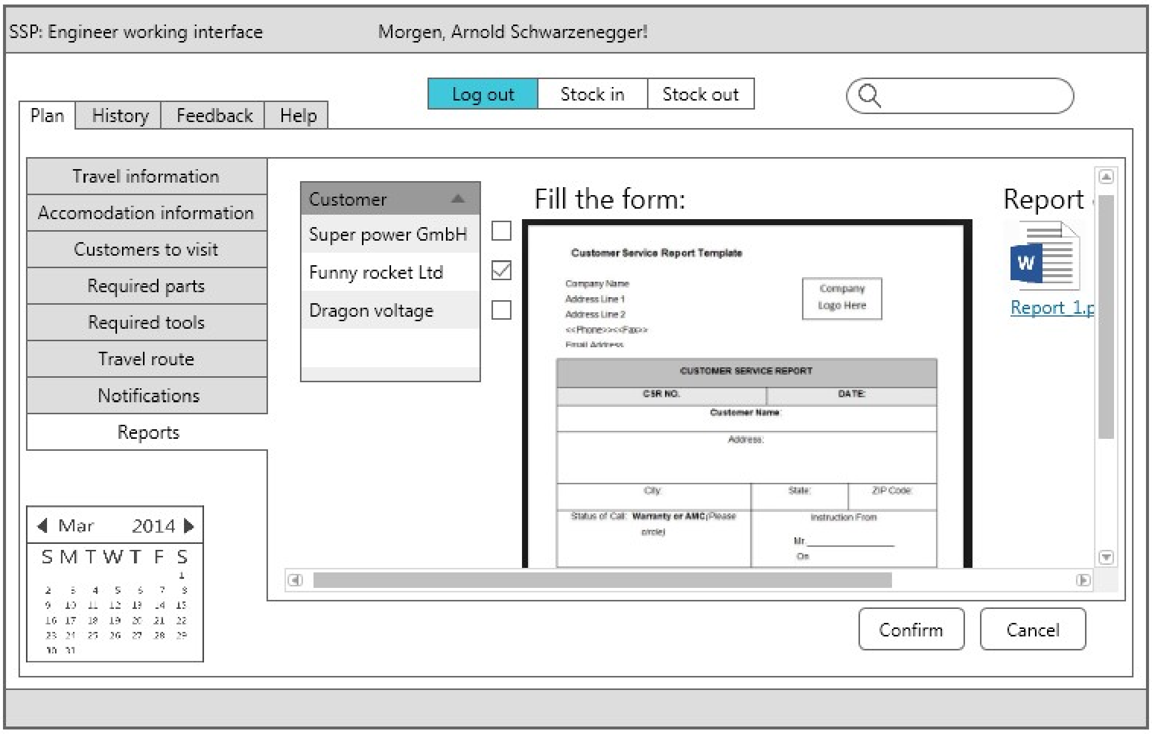
| Customer | Travel type | Departure time | Arrival time | Airport/rental compa |
|---|---|---|---|---|
| | By Plane | 01/04/19 07:30 | 01/04/19 09:00 | Tegel airport |
| Super power GmbH | Rental car | 01/04/19 07:30 | 01/04/19 09:00 | Alamo rental car |
| Funny rocket Ltd | By Plane | 01/04/19 11:20 | 01/04/19 13:00 | Lyon-Saint Exupéry A |
| Dragon voltage | Rental car | 01/04/19 07:30 | 01/04/19 09:00 | Hertz car rental |

Mar 2014

| S | M | T | W | T | F | S |
|---|---|---|---|---|---|---|
| | | | | | | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | | | | | |

**Figure 11.** SSP Service Engineer interface prototype 1.

As shown above during prototyping the widely used common solutions like v.tabs, tabs bar, checkboxes were used. The layout of SSP interface includes integrated possibility to log in and out to main company stock(storage). Button “Stock in” (internal login) is used in case, when employee is in corporate network. The button “stock out” (external login) is used in case, when employee is outside of the corporate network. The principle of login/logout procedure was the same as in old stock system. The tabs bar contains two important tabs: “Plan” and “History”. Clicking on “Plan” tab the user can see on the left side the list of required subcategories of service trip for the next week. Important information is stored on usual “excel”-type tables (This is done to make the transition of data from excels to SSP more easy). Rows of table “Customers to visit” equipped with checkboxes, selecting which the engineer can further confirm his choice by pressing button “Confirm” in the right side corner. Clicking on “Notification” tab redirecting the user to the field of listed ready service trip, concerning documents in pdf, word or other widely used formats to download them and print out if necessary. On next picture represented the “Reports” tab interface.



□ Super power GmbH
☑ Funny rocket Ltd
□ Dragon voltage

Customer Service Report Template

Company Name
Address Line 1
Address Line 2
<<Phone>><<Fax>>
Email Address

Company Logo Here

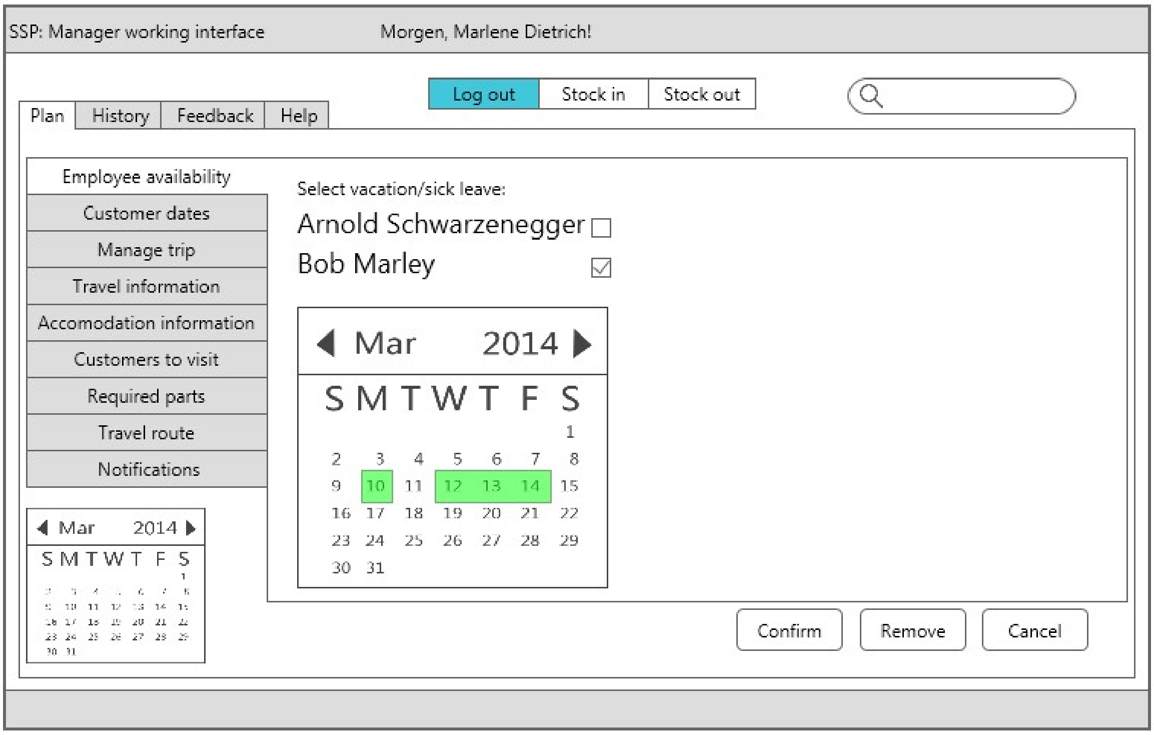
| CUSTOMER SERVICE REPORT | | |
|---|---|---|
| CSR NO. | DATE: | |
| Customer Name: | | |
| Address: | | |
| City: | State: | ZIP Code: |
| Status of Call: **Warranty or AMC** (Please circle) | Instruction From<br>Mr.____________<br>On | |

| Mar | | | | | | 2014 |
|---|---|---|---|---|---|---|
| S | M | T | W | T | F | S |
| | | | | | | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | | | | | |

**Figure 12.** SSP Service Engineer prototype interface 2.

In “Reports” tab engineer can select the report form of customer by making a tick in needed checkbox. After that, the proper report form will appear to be filled up. After the form is filled up by engineer it can be downloaded and signed by engineer and customer. Confirmation of used parts is done after all partnumbers are written in filled report form and the button “confirm” is pressed. The more precise description of interface graphical representation is represented in annexes.

The next picture represents the manager interface prototype:



Select vacation/sick leave:

Arnold Schwarzenegger □

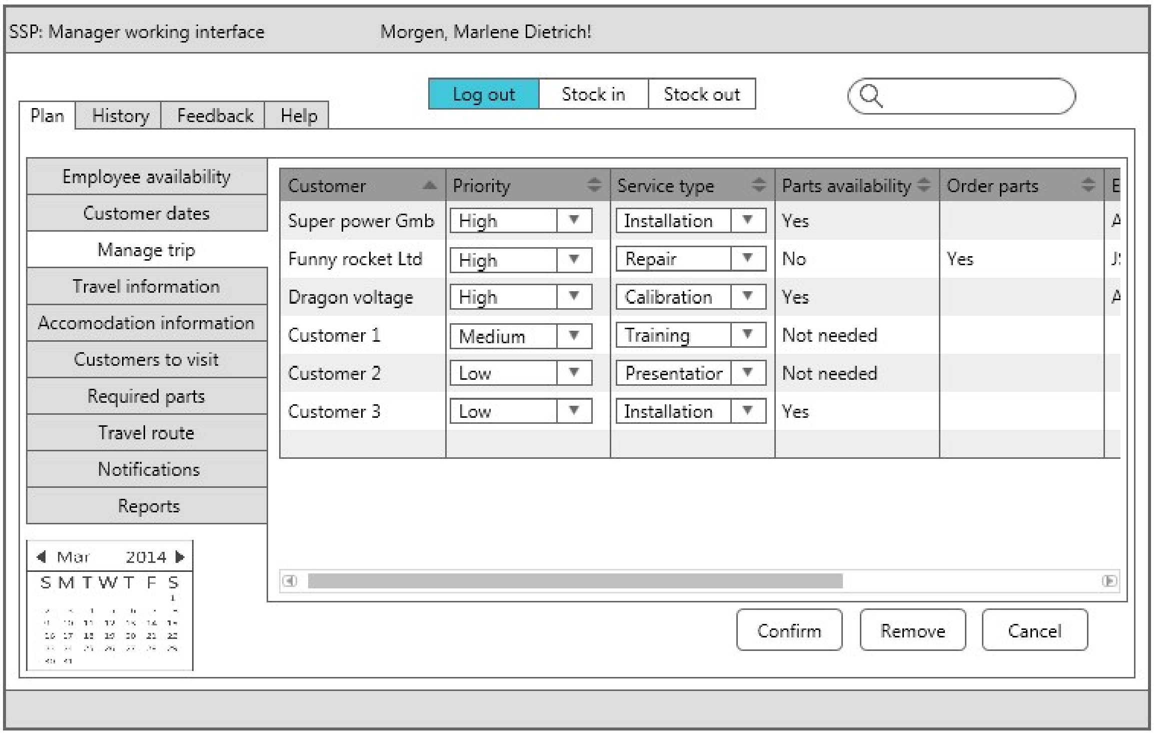
Bob Marley ☑

Mar 2014

| S | M | T | W | T | F | S |
|---|---|---|---|---|---|---|
|  |  |  |  |  |  | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 🟩 10 | 11 | 🟩 12 | 🟩 13 | 🟩 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 |  |  |  |  |  |

**Figure 13.** SSP Service manager interface prototype “Employee availability”.

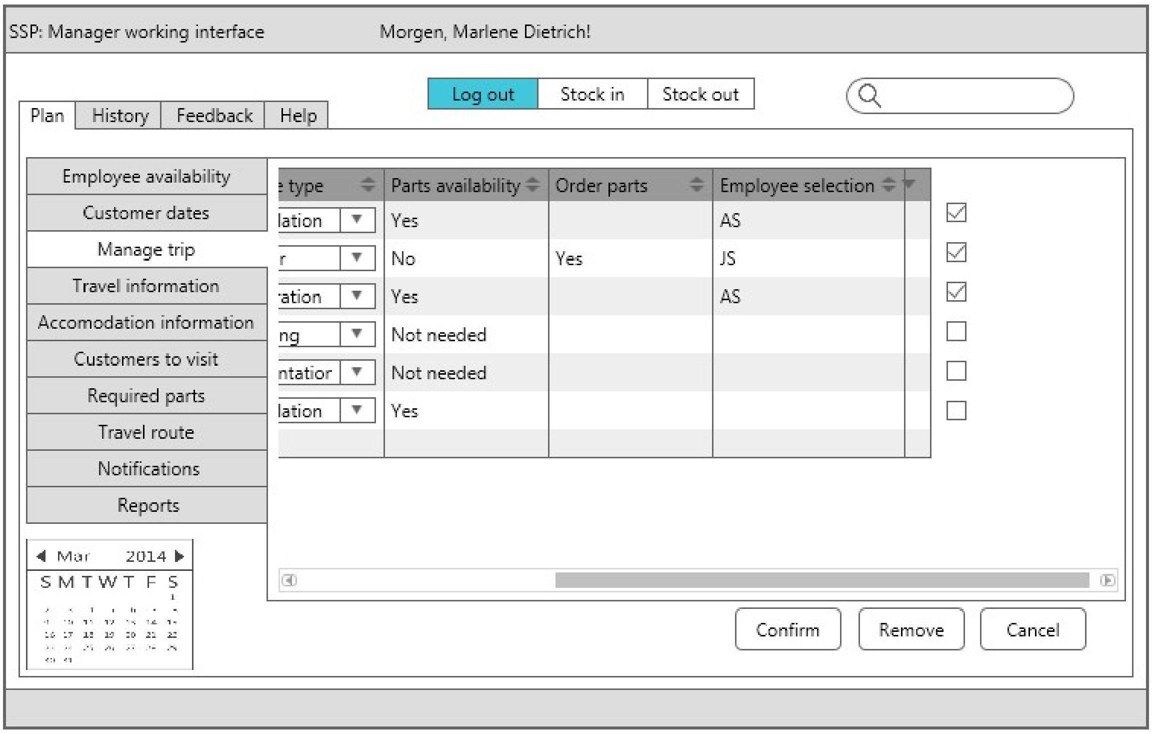
In comparison with engineer interface the functionality of manager interface is wider. In the picture above is shown the “employee availability” tab of manager interface. In presented interface the manager first can select the engineer by ticking proper checkbox. The calender for selected employee will appear. On the calender the manager can select engineer absence dates by simply clicking on appropriate date. The selection of service job priority and type can be done using the dropdown menu (see picture below).



| Customer | Priority | Service type | Parts availability | Order parts | E |
|---|---|---|---|---|---|
| Super power Gmb | High | Installation | Yes | | A |
| Funny rocket Ltd | High | Repair | No | Yes | J: |
| Dragon voltage | High | Calibration | Yes | | A |
| Customer 1 | Medium | Training | Not needed | | |
| Customer 2 | Low | Presentatior | Not needed | | |
| Customer 3 | Low | Installation | Yes | | |

**Figure 14.** SSP Service Manager interface prototype “Manage trips” 1.

All data containing tables are equipped with checkboxes (see picture below), selecting which can be confirmed also with the “Confirm” button or removed clicking on “Remove” button and after be filled with the data manually.



| e type | Parts availability | Order parts | Employee selection | |
|---|---|---|---|---|
| lation | Yes | | AS | ☑ |
| r | No | Yes | JS | ☑ |
| ration | Yes | | AS | ☑ |
| ng | Not needed | | | □ |
| ntatior | Not needed | | | □ |
| lation | Yes | | | □ |

**Figure 15.** SSP Service Manager interface prototype “Manage trips” 2.

After the confirmation, the SSP starts to refresh the overall routing and starts to create the necessary documents for the engineer in “Notification” tab. More functionality, concerning the prototype of the interface, can be found in annexes.

## 3.5 Construction

During Construction phase each microstep contains redevelopment of the architecture predefined in Elaboration phase or inserting new feature to the main architecture.

As new solution will operate with big amount of data and complex connections between systems with dynamically changed environment, there may appear a lot of restrictions. This is the reason, why under feature in current context we also mean add to SSP new restriction in functionality/change functionality. Next will be listed the possible restrictions to the system. The Construction phase can be the longest one in overall project, because there is no way to predict precisely such complex system and also will not be limited by current master´s thesis. Next will be represented three microsteps as an example of the overall phase procedure.

### 3.5.1 First update/microstep (Flight delays)

First restriction for planning procedure, which was not counted into main BPM model, was the possible flight delay. Flight delay may be longer then hours and in this case may affect all service trip. Below we can see the possible modification in BPM model. BMP model before modifications (The modified diagram will be presented in the next microstep):

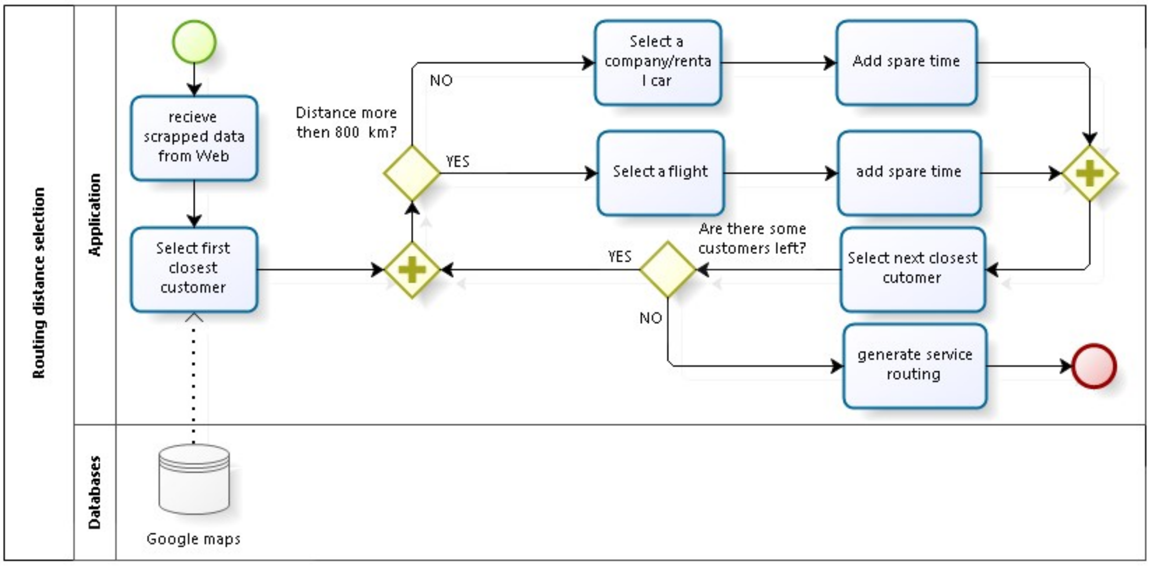

Routing distance selection
Application
Databases
recieve scrapped data from Web
Select first closest customer
Distance more then 800 km?
NO
YES
Select a company/rental car
Select a flight
Add spare time
add spare time
Are there some customers left?
YES
NO
Select next closest cutomer
generate service routing
Google maps

**Figure 16.** Initial routing distance selection BPM.

### 3.5.2 Second update/microstep (Traffic jam/car accidents)

If service engineers use a rental or company car to reach the customer and if the distance to the customer long enough, the probability of traffic jam or car accident might be high. In this case SSP should be able to take that fact into account. Below we can see possible modifications in BPM model

BPM model after modifications:

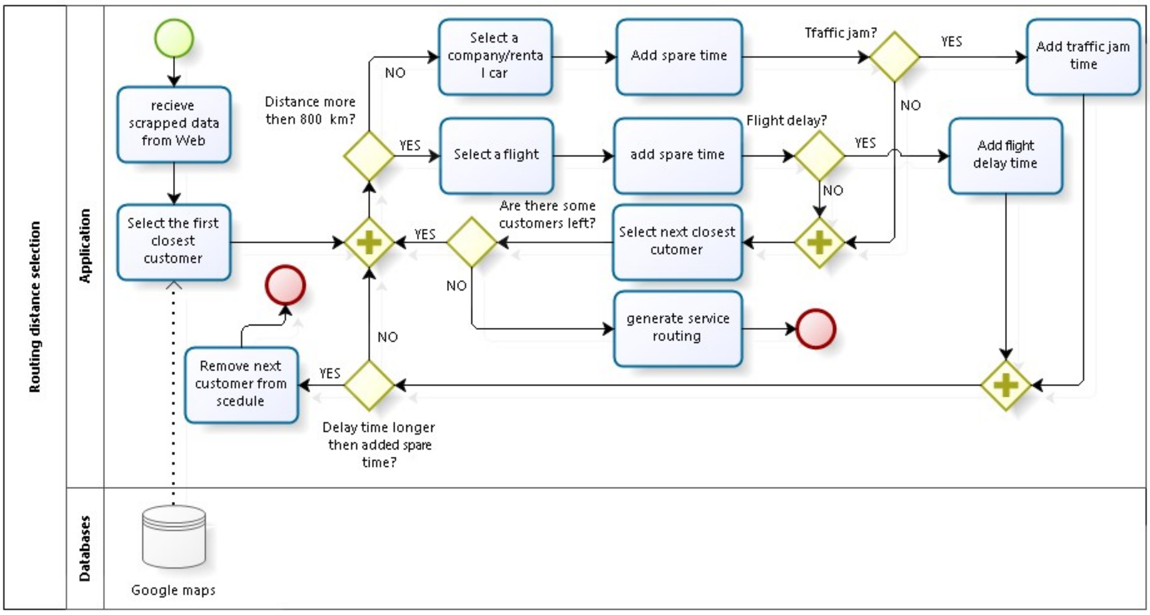

Routing distance selection
Application
Databases
recieve scrapped data from Web
Select the first closest customer
Distance more then 800 km?
NO
YES
Select a company/rental car
Select a flight
Add spare time
add spare time
Tfaffic jam?
YES
NO
Add traffic jam time
Flight delay?
YES
NO
Add flight delay time
Are there some customers left?
YES
NO
Select next closest cutomer
generate service routing
Remove next customer from scedule
YES
NO
Delay time longer then added spare time?
Google maps

**Figure 17.** Updated routing distance selection BPM.

### 3.5.3 Third update/microstep (Customer incomplete information about service)

There are quite frequent happened situations, in which customer do not provide the service center with full information about the problem before service. As a result, after the service engineer arrived to customer, there might happen, that not enough parts or tools were prepared. To eliminate the probability of happening such kind of situation the SSP should be able to predict the possible problem relying on service history of corresponding customer and the overall service history of certain product/model. Below we can see possible modifications in BPM model.

BPM model before modifications:

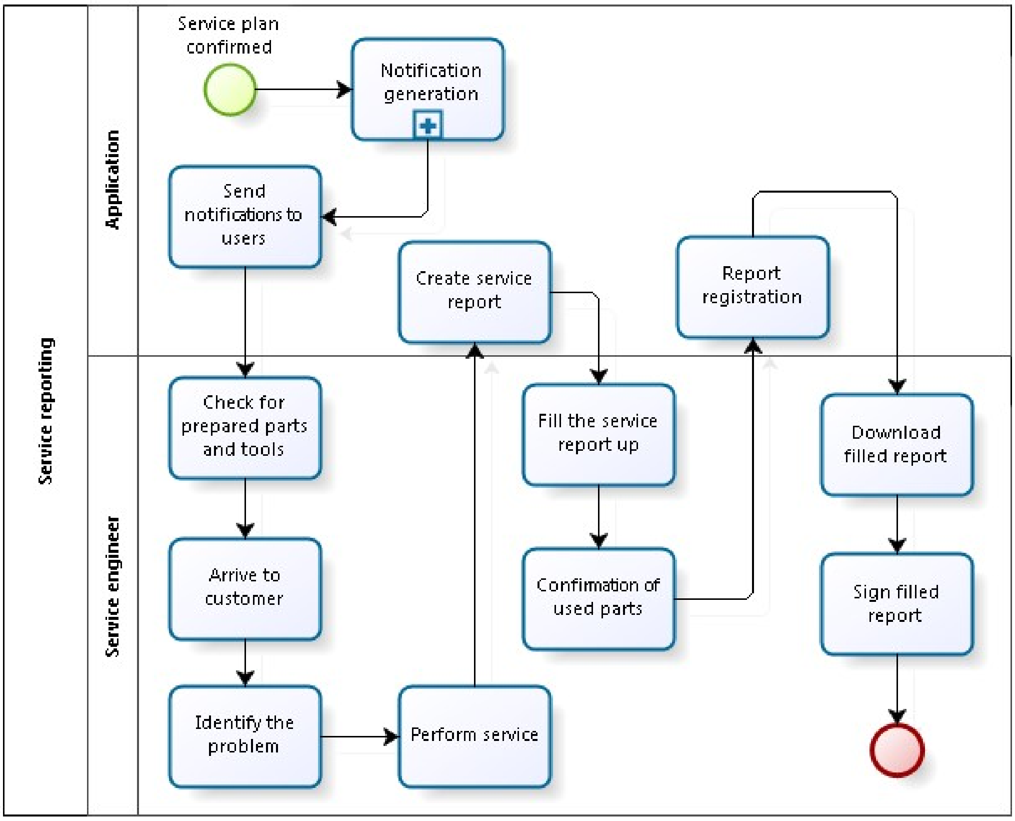


**Figure 18.** Initial service reporting BPM.

BPM model after modifications:

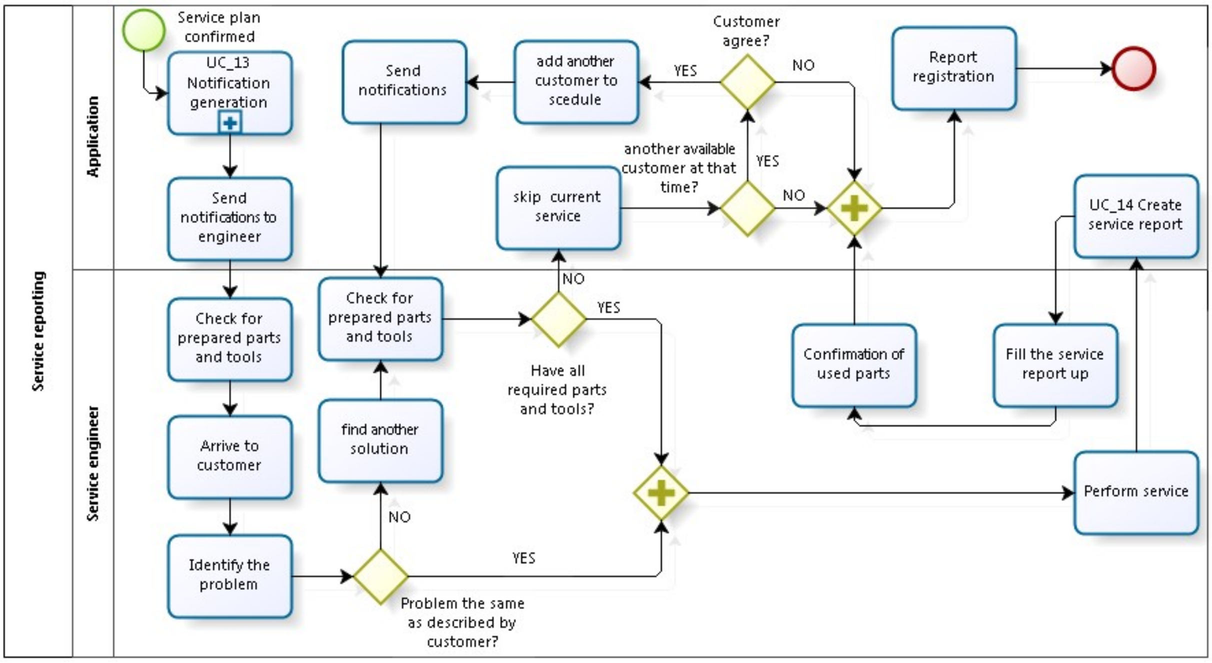

Service reporting
Application
Service engineer
Service plan confirmed
UC_13 Notification generation
Send notifications to engineer
Check for prepared parts and tools
Arrive to customer
Identify the problem
Problem the same as described by customer?
NO
YES
find another solution
Check for prepared parts and tools
Send notifications
add another customer to scedule
YES
Customer agree?
NO
skip current service
another available customer at that time?
YES
NO
NO
YES
Have all required parts and tools?
Report registration
Confirmation of used parts
Fill the service report up
UC_14 Create service report
Perform service

**Figure 19.** Updated service reporting BPM.

The prototype, after completed modifications described above, will have it initial form, because these modifications do not affect the prototype interfaces. The SSP will generate the report automatically in case of non-completed service and send it to manager.

Further modification and update procedure of the system will be done in next iterations. The rule “fail fast” should be related also to service jobs, because in this case the major of potential restrictions can be found earlier, and the number of iterations, before the project ends, will be limited and the final product will be released also earlier.

## 3.6 Transition

In Transition phase, the first iteration product version should be released, automatic test prepared. Tests performed to validate that user expectations are met. Taking into account that current master´s thesis author has only basic Python language skills, and the selected as a basis for GUI scripting language automation module AutoIt is written in C++, the selection of preferable programming language is under discussion. The transition phase is out of current master´s thesis scope.

In Transition phase the major emphasis is on code writing and automation test performing. This phase contains code writing and integration of system components with each other. Selection of programming language should rely on development team.

## 3.7 Chapter Summary

According OpenUP methodology the formalization level should be high at the beginning of the project. The initial state of business flow of the company before development starts was presented in BPM diagram. BPM diagram showed the significant work load on service manager during service planning. During development phase, the following actions were performed:

- User stories were collected during interview with engineers and manager.
- Determined new solution key components during project meeting discussion

During **Inception phase**:

- Business rules were defined
- Business roles and actors were defined
- Functional and non-functional requirements were defined
- New possible solution provided using BPMN notation.
- The scope of future solution was determined (Costs, Time, Resources and Risks)

During **Elaboration phase:**

- Use-case description based on user-stories were developed.
- Use-case model was created using StarUML freeware.
- SSP architecture component diagram was developed.
- SSP architecture deployment diagram was developed.
- SSP architecture sequence diagram was created.
- SSP architecture ERD model with semantics was created.
- SSP interface prototypes was developed.

During **Construction** phase, some examples of iteration procedure were presented on BPM diagrams, concerning exceptions, which were not taken into account during previous stages. As **Transition** phase is out of current master´s thesis scope, the only description of future actions like code/logic writing and test performing procedures was presented.

# 4 Security

The SSP application operates with no-sensitive information. All used information can be accessed freely by usual internet user. But there should be not forgotten, that even the used information is in free access for everyone, we use it to compose a plan, which has high influence on company expences. That means, that in some manner, there is a possible to create a potential harm script, which can replace input information by modified SSP logic with a wrong information and further, when data scrapping will be completed, the company will receive a totally wrong plan. To protect SSP application from unauthorized access to main logic code, the firewall was used and webpages, were SSP performs the information collection, uses HTTPS and SSL security sertificates. But even in case of using HTTPS and SSL certificates, there might be a possible threats:

- Malware

  After successful attack to web resource the malware can perform a SQL injection to infected webpage. In this case, the SSP application may select wrong link and download executable files to PC. To prevent this to happen SSP application compare the information, received during screen scraping with information, received and parced from HTML and XML script. If information match, the SSP check, what actions according to SSP logic, should be done with received information. The second method used, is to create in SSP logic restrictions for downloading only files with predefined extension (only pdf and doc for example)

- Phishing

  Phisher, after successful attack to resource, can insert to web page suspicious content (links), using the same symbols and sequence of them to make the application select replaced link or fill information to wrong fields. The principle of SSP behaviour in this case similar to behaviour in case of malware SQL injection. Additional to said above, the SSP application inserts into webbrowser only non-sensitive open information.

- Unauthorized access.

  The most dangerous situation may happen if third party performs an unauthorized access to company PC. In this case third party may steal customers personal information, which is stored on company server. In this case company uses known protection solutions against unathorized access like VPN connection and firewall.

The Security aspects will be discussed in next iteratons of the project and corrected according to changes in application architecture.

## 4.1 Chapter Summary

The SSP operates with non-sensitive and non-confidential information and to parce and collect information, SSP visits only webpages, which are using HTTPS and SSL sertificates. Protection against SQL injections and unauthorized access realized by combination of VPN and firewall solutions.

# 5 Recommendations for future work.

Current master´s thesis will be used as a basis for SSP application development for the inTEST Thermal Solutions GmbH. The OpenUP structure was implemented in current master´s thesis, to prepare the starting point of future project. The master´s thesis can be used as a first iteration of application development procedure. Taking into account, that the proposed solution of the master´s thesis have high amount of restrictions, the future development would have a lot of iterations. The interface of future solution might be tested for usability and functional performance. The structure of current solution was created in way to integrate already existing solutions like "AutoIt", „Selenium" and do as less modifications as possible to save time for development and concentrate on important aspects. The interface view and structure gives an opportunity to transfer already existing datatables in excel into new solution database with minimum amount of movements and modifications. Structure of current master´s thesis also was chosen to make it more simple to use in future development project realization.

# Summary

Based on the objectives and results obtained, the general conclusions of the thesis are as follows:

The routine daily operations with huge amout of continuously changing data in corresponding company is not efficient, if these operations are performed manually by current company limited labor resources. High valuable time, spent to plan the service jobs by the company service manager, does not give him a chance to concentrate on strategic marketing desicions. Also time, spent by the company engineer to prepare tools and parts, does not give him an opportunity to concentrate on technical aspect of the customer problem.

First output of current master´s thesis is successful methodology selection. A lot of exceptions in corresponding solution will exclude the „Waterfall“ variant from the scope. In the same time, the duration of current development will be long, because of the limited labor resources and that will exclude some iterative methodologies like „Scrum“. Current development has a high formalization level and precisely defined business rules and requirements at the initial phase of the project. The RUP methodology fullfills all requirements for current development, but it is to heavy for small companies and it is not open sourced. The alternative and the simplified form of RUP - the OpenUP open-sourced methodology was chosen for current master´s thesis development.

Second output of current master´s thesis is ready information architecture. According OpenUP methodology the formalization level should be high at the beginning of the project. The initial state of business flow of the company, before development starts, was presented in BPM diagram. BPM diagram showed the significant work load on service manager during service planning. During development phase following actions were performed:

- User stories were collected during interview with engineers and manager.

- There were determined new solution key components during project meeting discussion

During **Inception phase**:

- Business rules were defined
- Business roles and actors were defined
- Functional and non-functional requirements were defined
- New possible solution was provided using BPMN notation.
- The scope of future solution was determined (Costs, Time, Resources and Risks)

During **Elaboration phase:**

- Use-case description based on user-stories were developed.
- Use-case model was created using StarUML freeware.
- SSP architecture component diagram was developed.
- SSP architecture deployment diagram was developed.
- SSP architecture sequence diagram was created.
- SSP architecture ERD model with semantics was created.
- SSP interface prototypes was developed.

During **Construction** phase, an examples of iteration procedure were presented on BPM diagrams, concerning exceptions, which were not taken into account during previous stages. As **Transition** phase is out of current master´s thesis scope, the only description of future actions like code/logic writing and test performing procedures was presented. The master´s thesis pretend to be a first OpenUP iteration in future project.

The third output of current master´s thesis concerns security questions. The SSP operates with non-sensitive and non-confidential information and to parce and collect information SSP visits only webpages, which are using HTTPS and SSL sertificates. Protection against SQL injections and unauthorized access realized by combination of VPN and firewall solutions.

In conclusion, the company mentioned in this master´s thesis due to the nature of the service planning, forced to operate with big amount of data. It is obvious, that with the limited human resources and big amount of processed information, there is a need to create a efficient service plan, using computer science technologies, like virtual assistant or special application to process big data in fastest and efficient way. The modern RPA technology may be used also to accelerate the development. To rise the company revenue, the management board of last should be concentrated on strategic decisions. Other benefit of using new SSP solution is that there is not need anymore to doublecheck required tools and parts for the service or status of stock and spend for these actions more time. The SSP can be fast precise and informative.

The main and most important advantage after implementation of SSP solution might be more free time for management board to concentrate on strategic decisions on marketing planning. In this aspect, the revenue of the company can be higher, together with expansion of the company to other potential customer regions.

# References

[1] TTÜ, Lõputöö koostamise ja vormistamise juhend, Tallinn, 2018.

[2] Ian H. Witten, Eibe Frank, Mark A Hall, „Data Mining: Practical Machine Learning Tools and Techniques“ third edition, Burlington, USA, 2011.

[3] Ricardo Balduino, „Introduction to OpenUP (Open Unified Process), 2007.

[4] www.seleniumhq.org „Selenium-Web Browser Automation“.

[5] www.autoitscript.com, „AutoIt“.

[6] www.eclipse.org/epf/general/OpenUP.pdf, „Introduction to OpenUP“.

[7] Carvalho, Matheus C., “Practical Laboratory Automation: Made Easy with AutoIt”, 2016.

[8] Segev El, “Google and the Digital Divide: The Biases of Online Knowledge.”, Oxford, 2010.

[9] Cho, Junghoo, "Crawling the Web: Discovery and Maintenance of a Large-Scale Web Data", Ph.D. dissertation, Department of Computer Science, Stanford University, 2001.

[10] www.acunetix.com/websitesecurity/cross-site-scripting/, "Cross Site Scripting Attack"

[11] feross.org/html5-fullscreen-api-attack/, "Using the HTML5 Fullscreen API for Phishing Attacks”.

[12] John C. Mitchell, "Browser Security Model", Stanford, 2010.

[13] "OWASP Top 10 - 2017: The Ten Most Critical Web Application Security Risks". Open Web Application Security Project, 2017.

[14] Microsoft, docs.microsoft.com, "SQL Injection", 2017.

[15] Чубукова И. А., „Data Mining: учебное пособие. — М.: Интернет-университет информационных технологий: БИНОМ: Лаборатория знаний, 2006

[16] Alexander S., Brian D., „Intelligent agents as innovations“, London, 2004.

[17] Liman Mao, „Web-based Information System for Land Management“, Master Thesis, Alberta, 2005.

[18] Jeffrey M. Bradshaw, „An Introduction to Software Agents“, USA, 1997.

[19] Andreas Mehlführer, „Web Scraping A Tool Evaluation“ Diplomarbeit, Wien, 2009.

[20] Jeffrey K. Hirschey, „Symbiotic Relationships: Pragmatic Acceptance of Data Scraping“, 29 Berkeley Tech L.J., 2014.

[21] А.С.Передерей, “Проектирование унифицированного процесса для компании RedLine Group”, Томск, 2010.

[22] Bolin, M., M. Webber, P. Rha, T. Wilson, and R. C. Miller. „Automation and customization of rendered web pages“, 2005.

[23] Potter, R. L. „Pixel data access: interprocess communication in the user interface for end-user programming and graphical macros“, Ph. D. thesis , USA, 1999.

[24] Tom Yeh, „Sikuli: Using GUI Screenshots for Search and Automation“, USA, 2009.

# Appendix 1 – Use-case “Routing distance selection” description.

**Table 1.** Use-case **“**Routing distance selection” description.

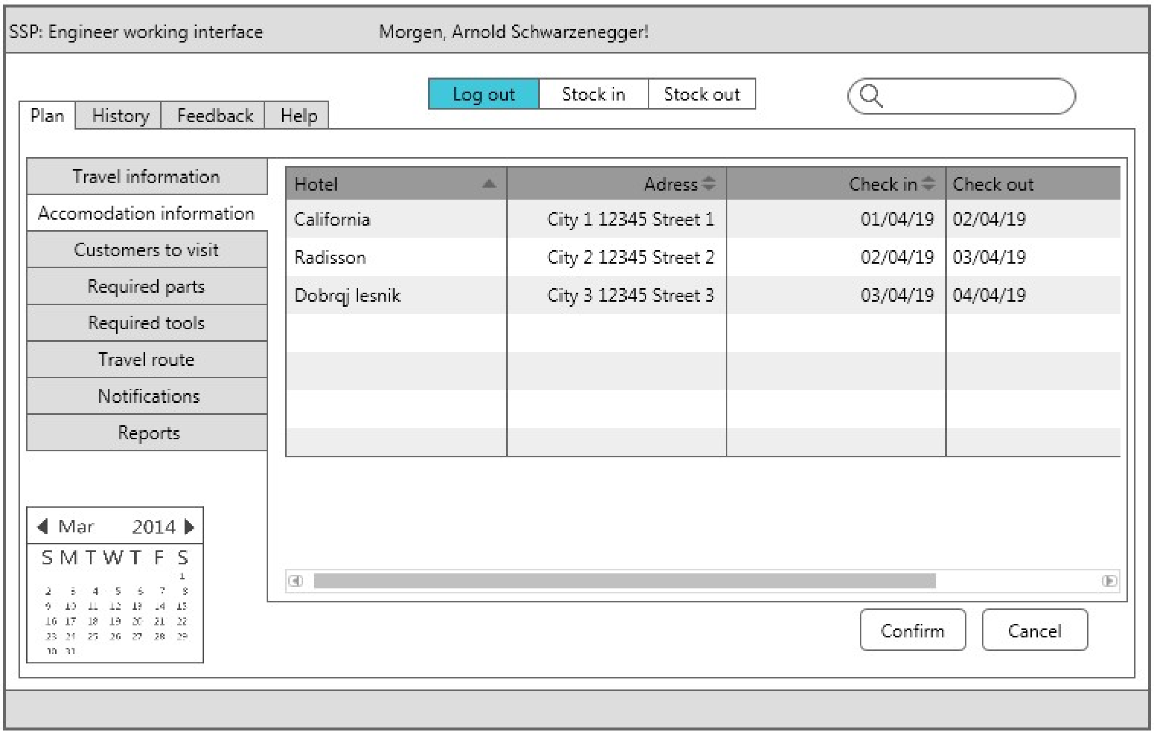
| | |
|---|---|
| Use case name | Routing distance selection |
| ID | UC_ID_21 |
| Description | Application establish the shortest possible service trip. |
| Actors | Application |
| Triggers | |
| Preconditions | Applcation installed, all preferable dates by customer selected, all dates with available employees are selected by service manager, scraped data received |
| Postconditions | Service trip created. |
| Main Course | 1. Receive scraped data from websites.<br>2. Select from scraped data list the closest customer.<br>3. Calculate the distance.<br>4. Select a flight, if distance is more than 800 km.<br>5. Add spare time by default(for flights 3 hours)<br>7. Check, is there delay in flight?<br>8. Select another closest customer, in case, if there is no delay.<br>9. Check for next available customer?<br>10. Repeat main course in case, if next customer exists. |
| Alternate Courses | 1. Receive scraped data from websites.<br>2. Select from scraped data list the closest customer.<br>3. Calculate the distance.<br>4. Select a rental car company, if the distance less than 800 km<br>5. Add spare time by default(for rental car 2 hours)<br>7. Check, is there delay traffic jam?<br>8. Select another closest customer in case if there is no traffic jam.<br>9. Check for next available customer?<br>10. Repeat main course in case, if next customer exists. |
| Exceptions | To be considered |
| | |

# Appendix 2 – Use-case “Service reporting” description.

**Table 1.** Use-case “Service reporting” description.

| | |
|---|---|
| Use case name | Service reporting |
| ID | UC_ID_22 |
| Description | Application manages service reporting procedure. |
| Actors | Application, Service engineer |
| Triggers | |
| Preconditions | Service plan confirmed |
| Postconditions | Completed service, service report registration in database, service history updated. |
| Main Course | 1. Ready service notifications sent to service engineer.<br>2. The service engineer check required tools and parts.<br>3. The service engineer arrives to the customer site.<br>4. The service engineer finds out, if the problem is the same as described by the customer?<br>5. Engineer performs the service, in case if problem is the same as described.<br>6. Application creates service report for current service.<br>7. The service engineer fills up the report.<br>8. Engineer confirms the used parts.<br>9. Application registers ready report in the system. |
| Alternate Courses | 1. Ready service notifications sent to service engineer<br>2. The service engineer check required tools and parts<br>3. The service engineer arrives to the customer site.<br>4. The service engineer finds out, if the problem is the same as described by the customer?<br>5. The service engineer tries to find a solution, in case if the problem is not the same as described.<br>6. The service engineer check for required tools and parts, in case if a new solution was found.<br>5. Engineer performs the service, in case if all required tools and parts were found.<br>6. Application creates service report for current service.<br>7. The service engineer fills up the report.<br>8. Engineer confirms the used parts.<br>9. Application registers ready report in the system. |
| Exceptions | To be considered |
| | |

# Appendix 3 – Engineer working interface „Accomodation information“ tab view.

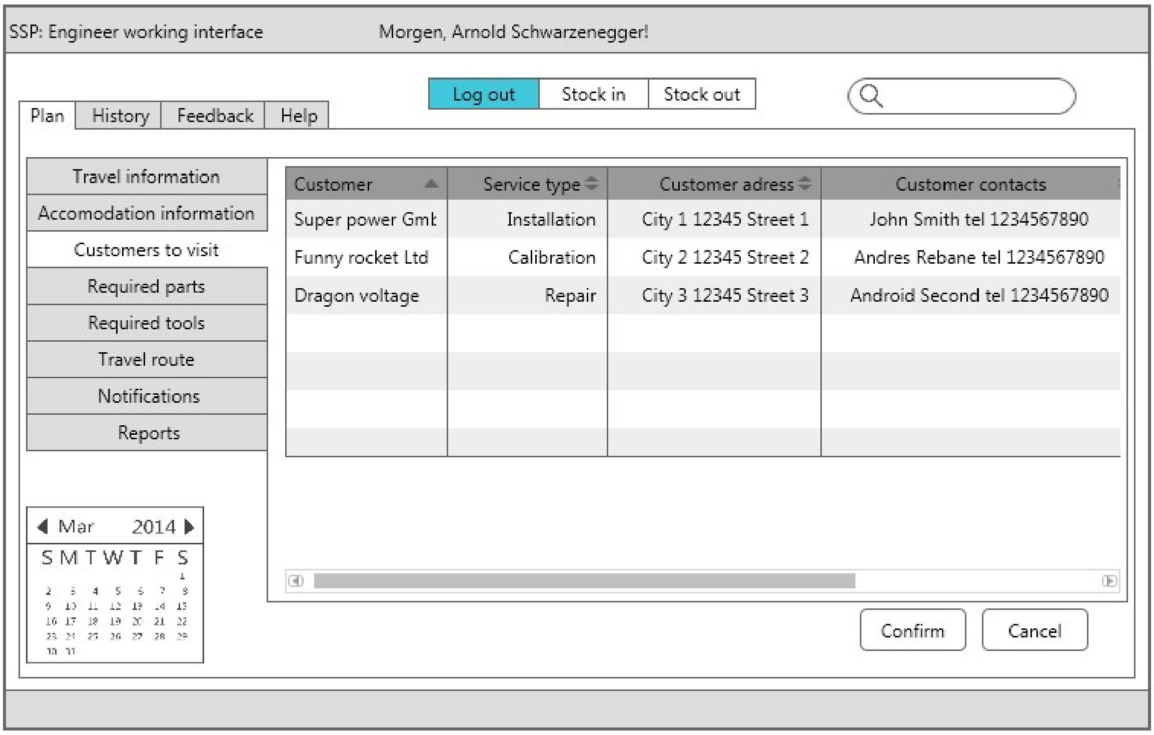


| Hotel | Adress | Check in | Check out |
|---|---|---|---|
| California | City 1 12345 Street 1 | 01/04/19 | 02/04/19 |
| Radisson | City 2 12345 Street 2 | 02/04/19 | 03/04/19 |
| Dobrqj lesnik | City 3 12345 Street 3 | 03/04/19 | 04/04/19 |

Figure 1. Engineer working interface „Accomodation information“ tab view.

Selecting „Accomodation information“ tab the service engineer can see all required information. The information presened in table form to make it easier to transfer data to excel format if required.

# Appendix 4 – Engineer working interface „Customers to visit“ tab view.



| Customer | Service type | Customer adress | Customer contacts |
|---|---|---|---|
| Super power Gmb | Installation | City 1 12345 Street 1 | John Smith tel 1234567890 |
| Funny rocket Ltd | Calibration | City 2 12345 Street 2 | Andres Rebane tel 1234567890 |
| Dragon voltage | Repair | City 3 12345 Street 3 | Android Second tel 1234567890 |

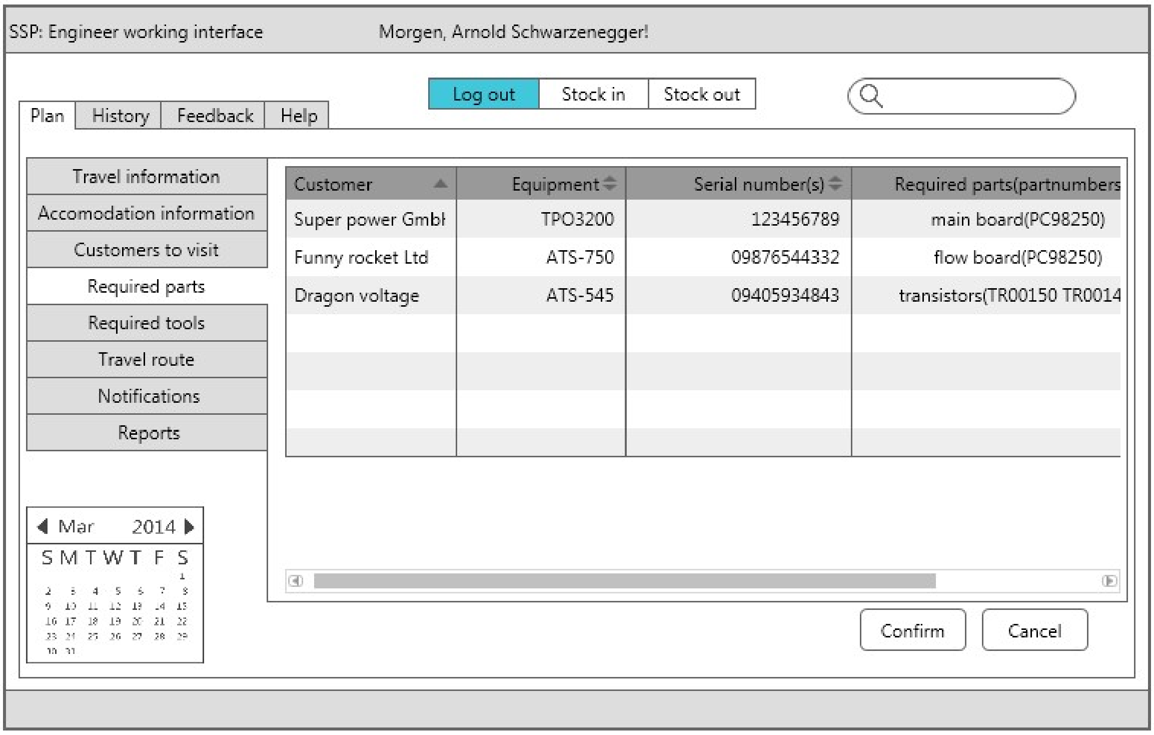
Mar 2014

| S | M | T | W | T | F | S |
|---|---|---|---|---|---|---|
| | | | | | | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | | | | | |

Figure 2. Engineer working interface „Customers to visit“ tab view.

Selecting „Customers to visit“ tab the service engineer can see all required information. The information presened in table form to make it easier to transfer data to excel format if required.

# Appendix 5 – Interface prototype „Required parts“ tab view.



| Customer | Equipment | Serial number(s) | Required parts(partnumbers |
|---|---|---|---|
| Super power Gmbł | TPO3200 | 123456789 | main board(PC98250) |
| Funny rocket Ltd | ATS-750 | 09876544332 | flow board(PC98250) |
| Dragon voltage | ATS-545 | 09405934843 | transistors(TR00150 TR0014 |

Figure 3. Interface prototype „Required parts“ tab view.

Selecting „Required parts“ tab service engineer can see all required information. Engineer can see the location of the components on stock, serial number, partnumer and the equipment type. The information presened in table form to make it easier to transfer data to excel format if required.

# Appendix 6 – Interface prototype „Travel route“ tab view.

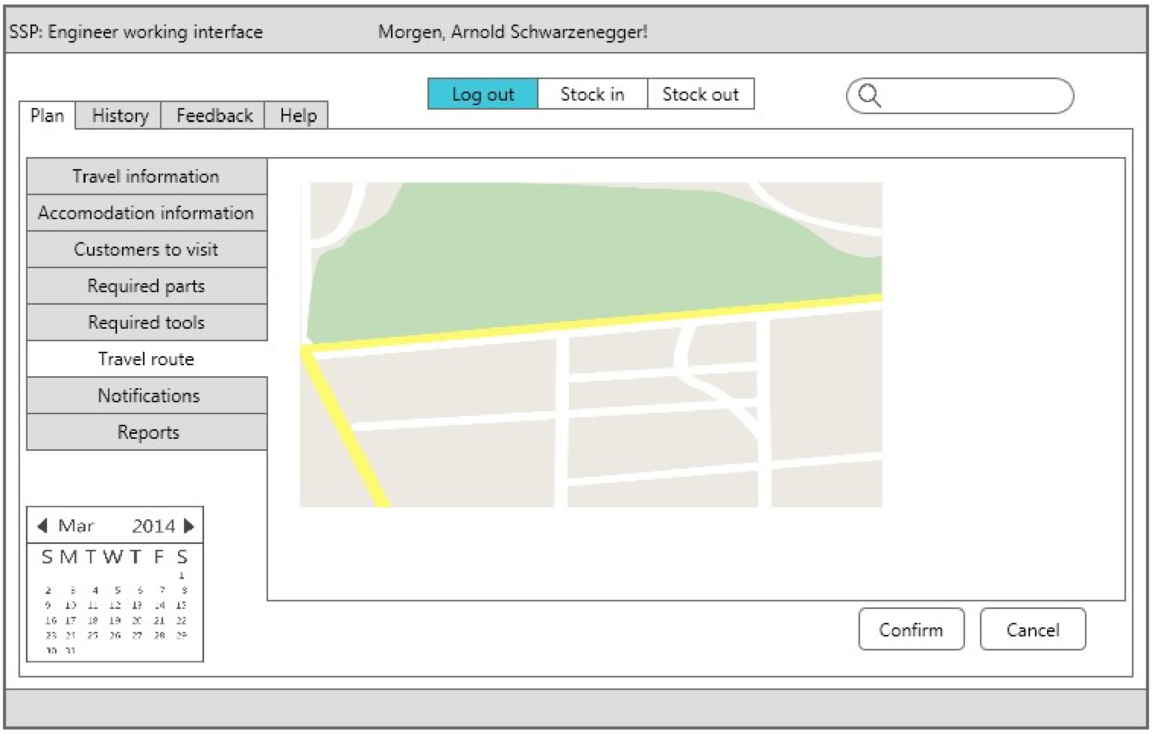
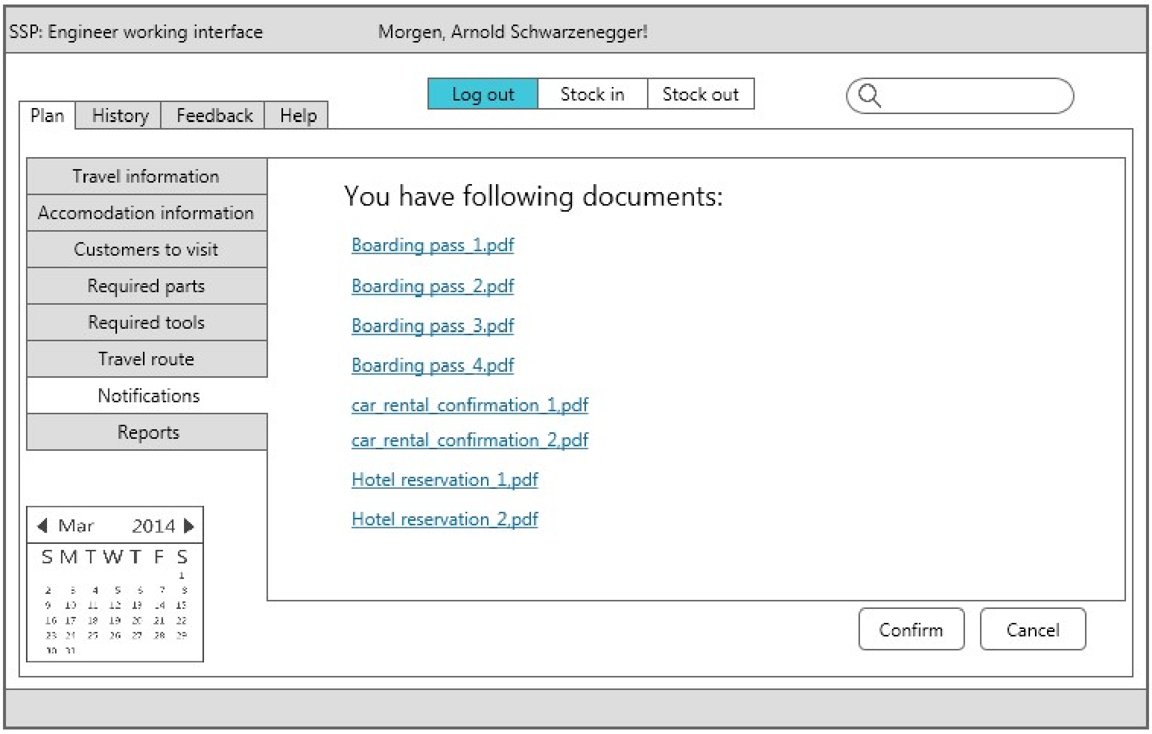


Figure 4. Interface prototype „Travel route“ tab view.

Selecting „Travel route“ tab service engineer can see the map with all travel routings and locations of all customers, airports and car rental companies. Engineer can enlarge or minimize the map to navigate himself.

# Appendix 7 – Interface prototype „Notification“ tab view.



SSP: Engineer working interface

Morgen, Arnold Schwarzenegger!

You have following documents:

Boarding pass 1.pdf

Boarding pass 2.pdf

Boarding pass 3.pdf

Boarding pass 4.pdf

car rental confirmation 1.pdf

car rental confirmation 2.pdf

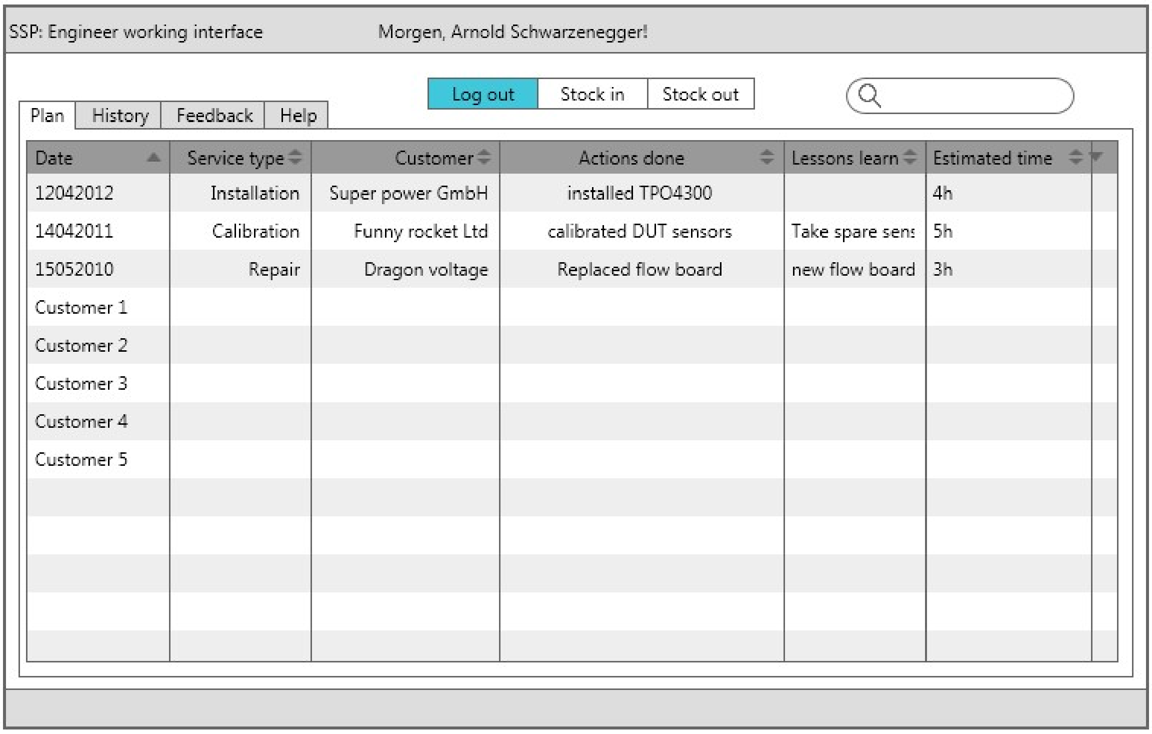
Hotel reservation 1.pdf

Hotel reservation 2.pdf

Figure 5. Interface prototype „Notification“ tab view.

Selecting „Notification“ tab engineer can find all documents attached to week service plan. From this field is possible to download all needed papers for hotel reservation, boarding passes and car rental confirmations.

# Appendix 8 – Interface prototype „History“ tab view.



SSP: Engineer working interface

Morgen, Arnold Schwarzenegger!

| Date | Service type | Customer | Actions done | Lessons learn | Estimated time |
|---|---|---|---|---|---|
| 12042012 | Installation | Super power GmbH | installed TPO4300 | | 4h |
| 14042011 | Calibration | Funny rocket Ltd | calibrated DUT sensors | Take spare sens | 5h |
| 15052010 | Repair | Dragon voltage | Replaced flow board | new flow board | 3h |
| Customer 1 | | | | | |
| Customer 2 | | | | | |
| Customer 3 | | | | | |
| Customer 4 | | | | | |
| Customer 5 | | | | | |

Figure 6. Interface prototype „History“ tab view.

Selecting „History“ tab from upper „Plan, History, Feedback, Help“ menu service engineer can see information about done service jobs. The information stored in corresponding table can be used before preparing to servce of already known machine. It can be helpful to read the reasons of misfunction and the methods used to fix the problem. The information presened in table form to make it easier to transfer data to excel format if required.

# Appendix 9 – Interface prototype „Customer dates“ manager tab view.

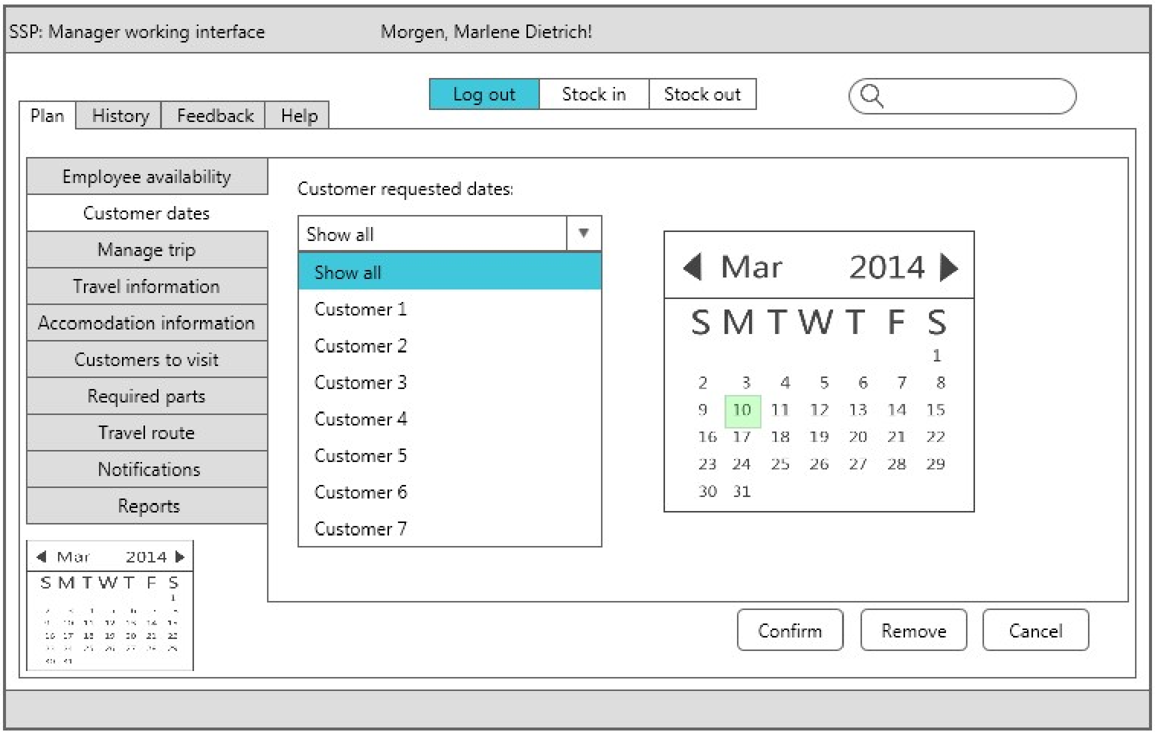
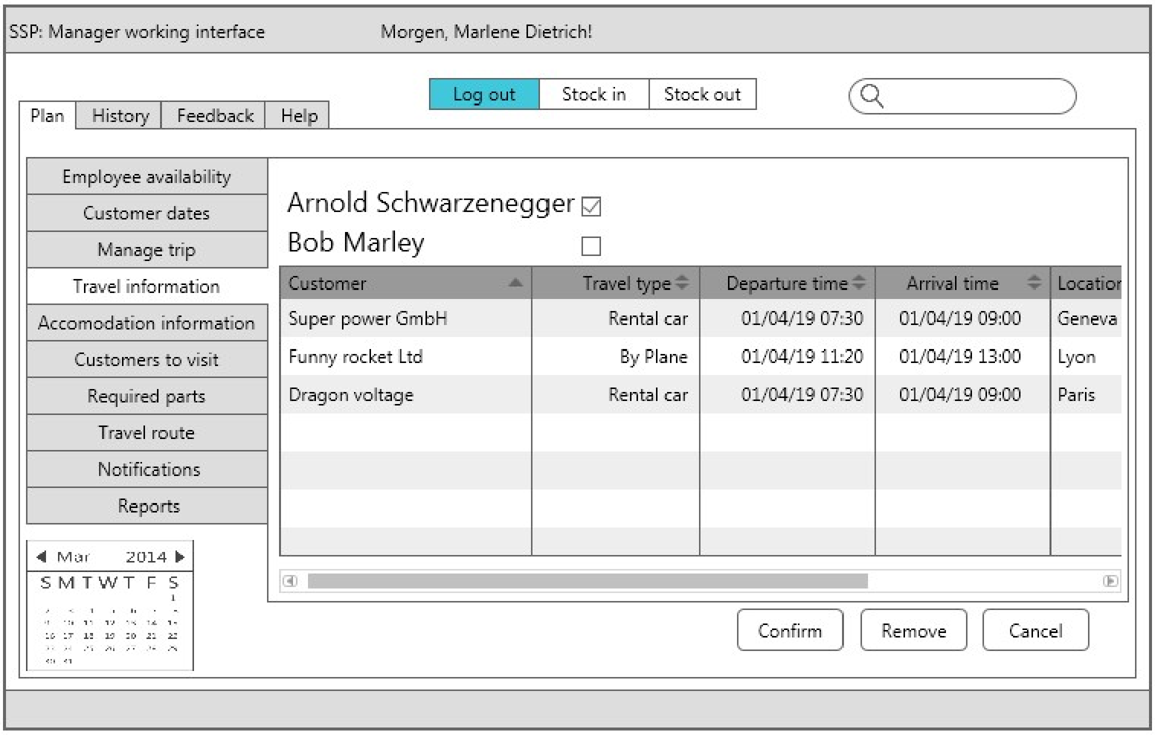


Figure 7. Interface prototype „Customer dates“ manager tab view.

Selecting „Customer dates“ tab service manager can chose from dropdown menu required customer and see the service date selected by the customer. Clicking on „Confirm“ button service manager can confirm the date or refuse pressing the „Remove“ button and select manualy requied service date.

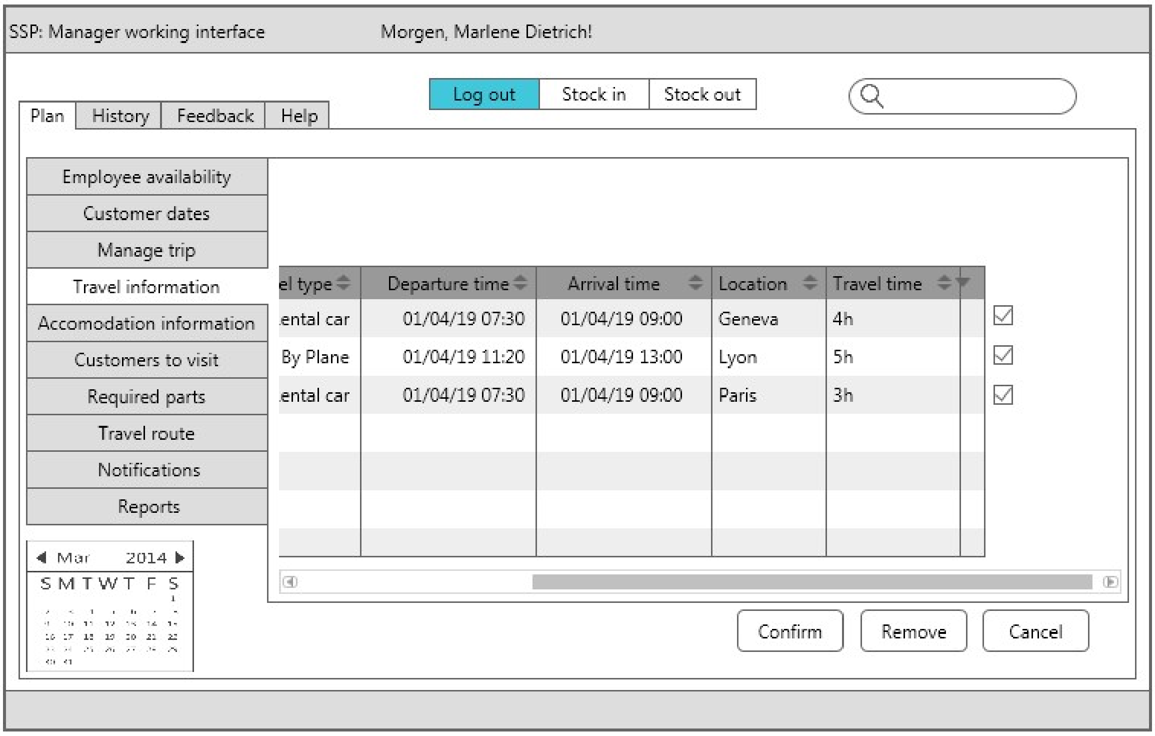
## Appendix 10 – Interface prototype „Travel information“ manager tab view.



Arnold Schwarzenegger ☑

Bob Marley □

| Customer | Travel type | Departure time | Arrival time | Locatior |
|---|---|---|---|---|
| Super power GmbH | Rental car | 01/04/19 07:30 | 01/04/19 09:00 | Geneva |
| Funny rocket Ltd | By Plane | 01/04/19 11:20 | 01/04/19 13:00 | Lyon |
| Dragon voltage | Rental car | 01/04/19 07:30 | 01/04/19 09:00 | Paris |

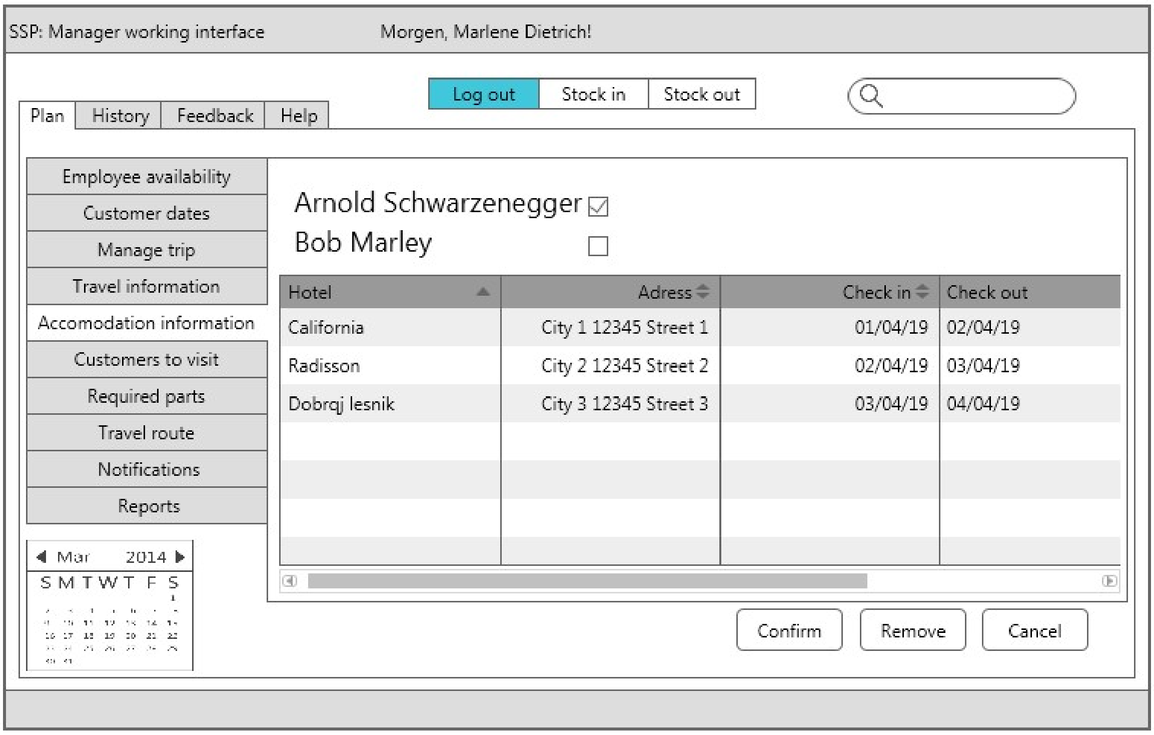


| el type | Departure time | Arrival time | Location | Travel time | |
|---|---|---|---|---|---|
| ental car | 01/04/19 07:30 | 01/04/19 09:00 | Geneva | 4h | ☑ |
| By Plane | 01/04/19 11:20 | 01/04/19 13:00 | Lyon | 5h | ☑ |
| ental car | 01/04/19 07:30 | 01/04/19 09:00 | Paris | 3h | ☑ |

Figure 8. Interface prototype „Travel information“ manager tab view.

Selecting „Travel information“ tab service manager can select first the service engineer making a tip on corresponding checkbox and see all bussines travel locations of this engineer which can be confirmed with selection of checkbox near the travel location row and pressing the „Confirm“ button or reject pressing the „Remove“ button and manually filling the empty row of removed travel location point. All information is presented in table form to make it easier to transfer required data to and from excel file.

# Appendix 11 – Manager interface prototype „Accomodation information“ tab view.



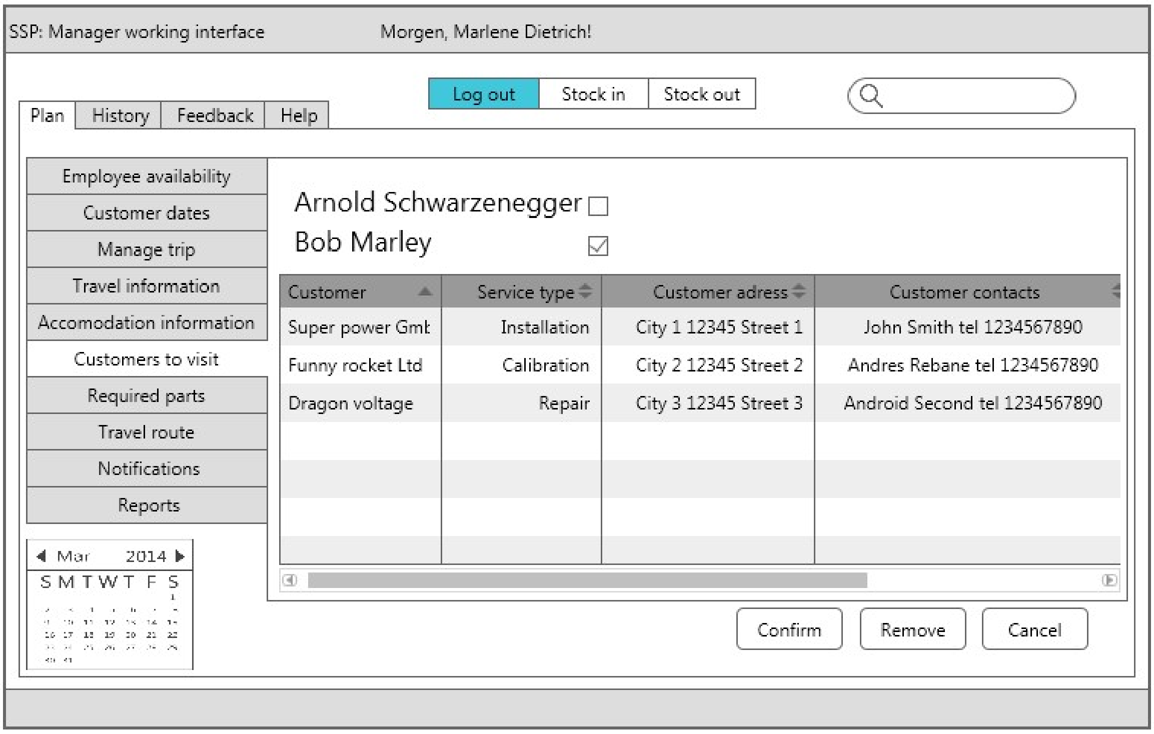
Arnold Schwarzenegger ☑
Bob Marley ☐

| Hotel | Adress | Check in | Check out |
|---|---|---|---|
| California | City 1 12345 Street 1 | 01/04/19 | 02/04/19 |
| Radisson | City 2 12345 Street 2 | 02/04/19 | 03/04/19 |
| Dobrqj lesnik | City 3 12345 Street 3 | 03/04/19 | 04/04/19 |

Figure 9. Manager interface prototype „Accomodation information“ tab view.

Selecting „Accomodation information“ tab service manager can select first the service engineer making a tip on corresponding checkbox and see all accomodation locations of this engineer which can be confirmed with selection of checkbox near the accomodation location row and pressing the „Confirm“ button or reject pressing the „Remove“ button and manually filling the empty row of removed accomodation location point. All information is presented in table form to make it easier to transfer required data to and from excel file.

# Appendix 12 – Manager interface prototype „Customers to visit“ tab view.



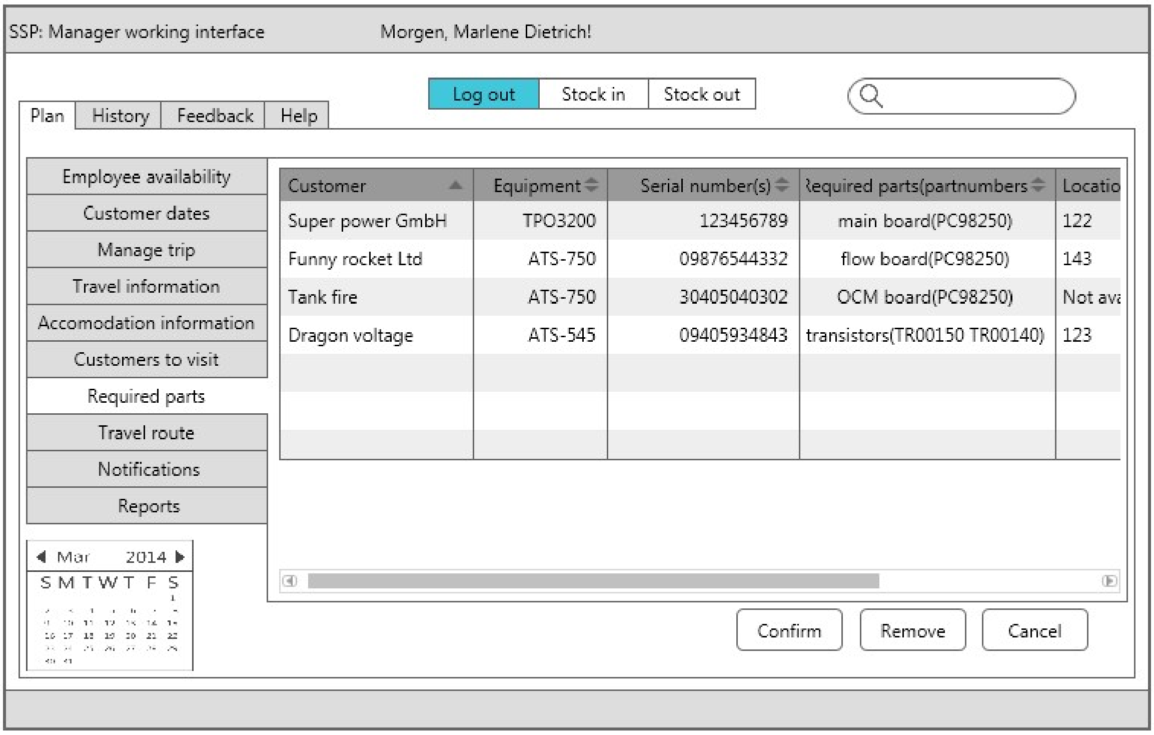
Arnold Schwarzenegger □
Bob Marley ☑

| Customer | Service type | Customer adress | Customer contacts |
|---|---|---|---|
| Super power Gmt | Installation | City 1 12345 Street 1 | John Smith tel 1234567890 |
| Funny rocket Ltd | Calibration | City 2 12345 Street 2 | Andres Rebane tel 1234567890 |
| Dragon voltage | Repair | City 3 12345 Street 3 | Android Second tel 1234567890 |

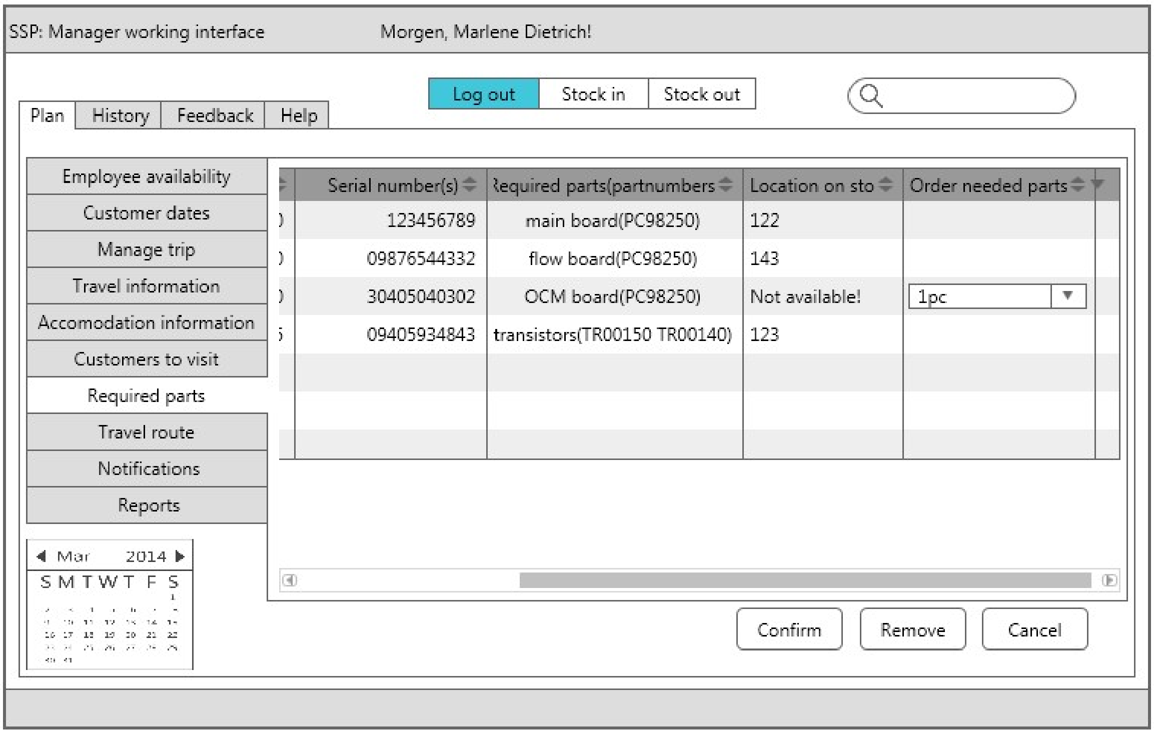
Figure 10. Manager interface prototype „Customers to visit“ tab view.

Selecting „Customers to visit“ tab service manager can select first the service engineer making a tip on corresponding checkbox and see all customers of this engineer which can be confirmed with selection of checkbox near the customer row and pressing the „Confirm“ button or reject pressing the „Remove“ button and manually filling the empty row of removed customer. All information is presented in table form to make it easier to transfer required data to and from excel file.

# Appendix 13 – Manager interface prototype „Required parts“ tab view.



| Customer | Equipment | Serial number(s) | Required parts(partnumbers | Locatio |
|---|---|---|---|---|
| Super power GmbH | TPO3200 | 123456789 | main board(PC98250) | 122 |
| Funny rocket Ltd | ATS-750 | 09876544332 | flow board(PC98250) | 143 |
| Tank fire | ATS-750 | 30405040302 | OCM board(PC98250) | Not ava |
| Dragon voltage | ATS-545 | 09405934843 | transistors(TR00150 TR00140) | 123 |

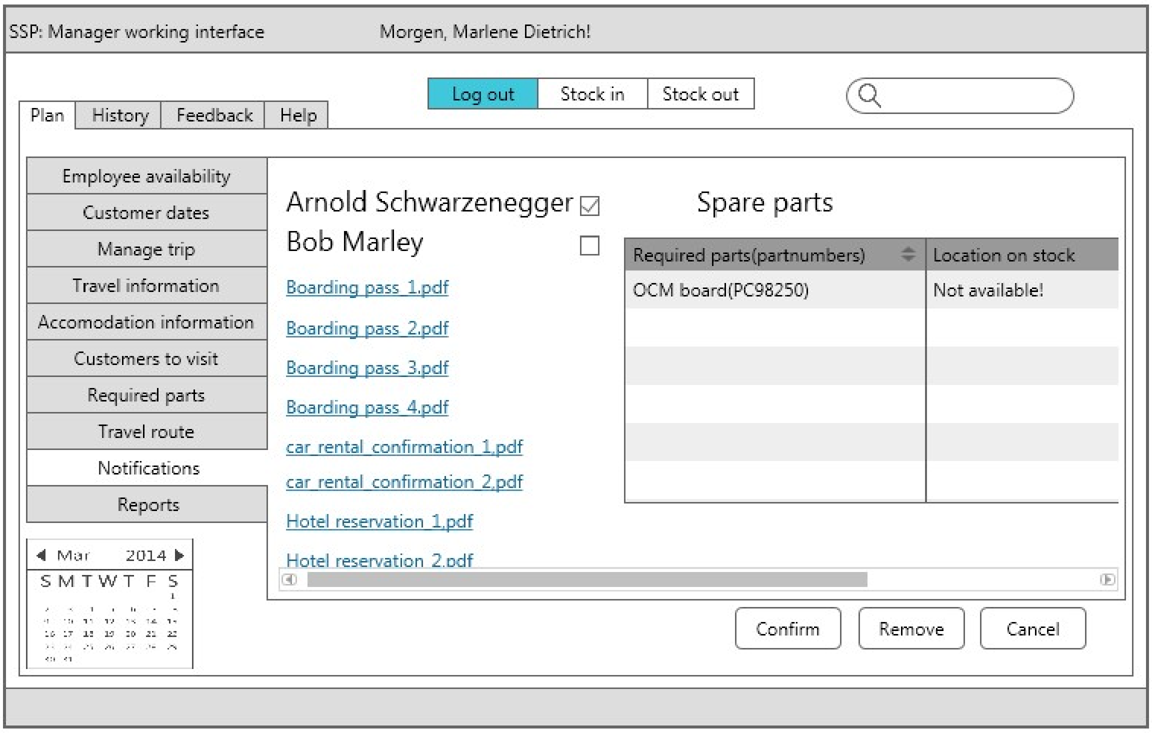


| | Serial number(s) | Required parts(partnumbers | Location on sto | Order needed parts |
|---|---|---|---|---|
| ) | 123456789 | main board(PC98250) | 122 | |
| ) | 09876544332 | flow board(PC98250) | 143 | |
| ) | 30405040302 | OCM board(PC98250) | Not available! | 1pc ▼ |
| 5 | 09405934843 | transistors(TR00150 TR00140) | 123 | |

Figure 11. Manager interface prototype „Required parts“ tab view.

Selecting „Required parts“ tab service manager can see required partslist for all service jobs. There is possible to read out the name, serial number, part number, location on stock from the presented table and order needed parts if some of them is not available, selecting from dropdown menu required amount of ordered parts.

# Appendix 14 – Interface prototype „Notifications“ tab view.



Arnold Schwarzenegger ☑

Bob Marley □

Boarding pass 1.pdf

Boarding pass 2.pdf

Boarding pass 3.pdf

Boarding pass 4.pdf

car rental confirmation 1.pdf

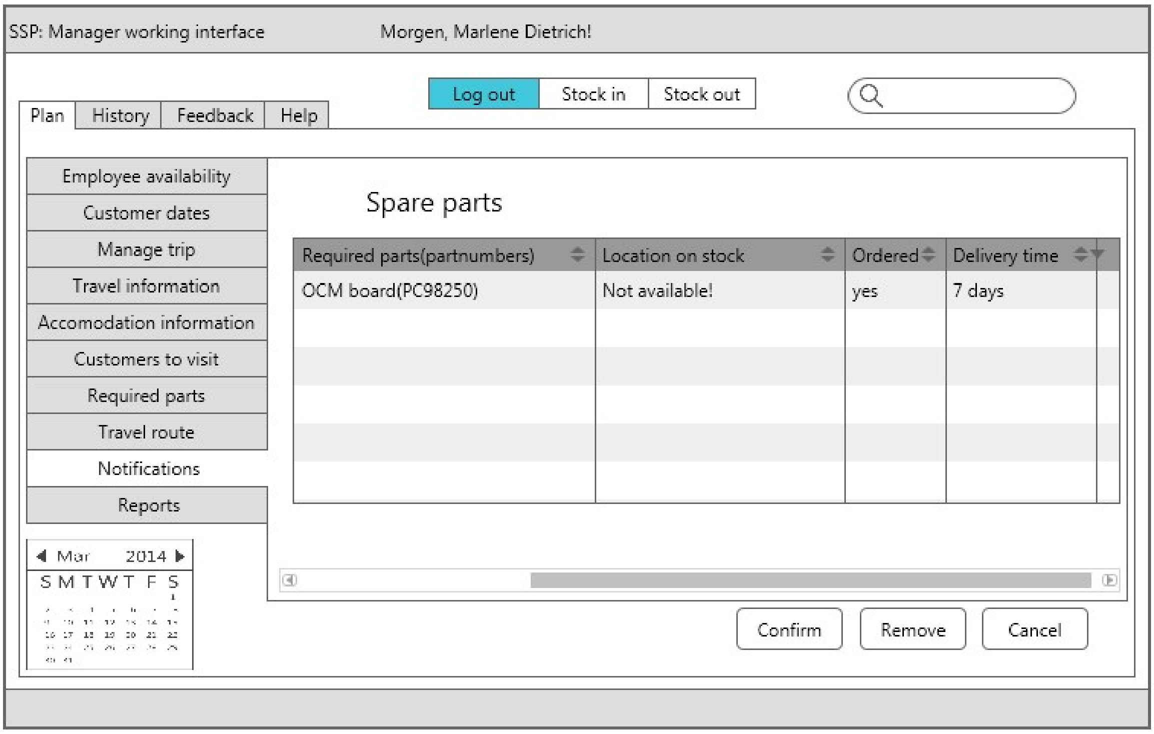
car rental confirmation 2.pdf

Hotel reservation 1.pdf

Hotel reservation 2.pdf

Spare parts

| Required parts(partnumbers) | Location on stock |
|---|---|
| OCM board(PC98250) | Not available! |



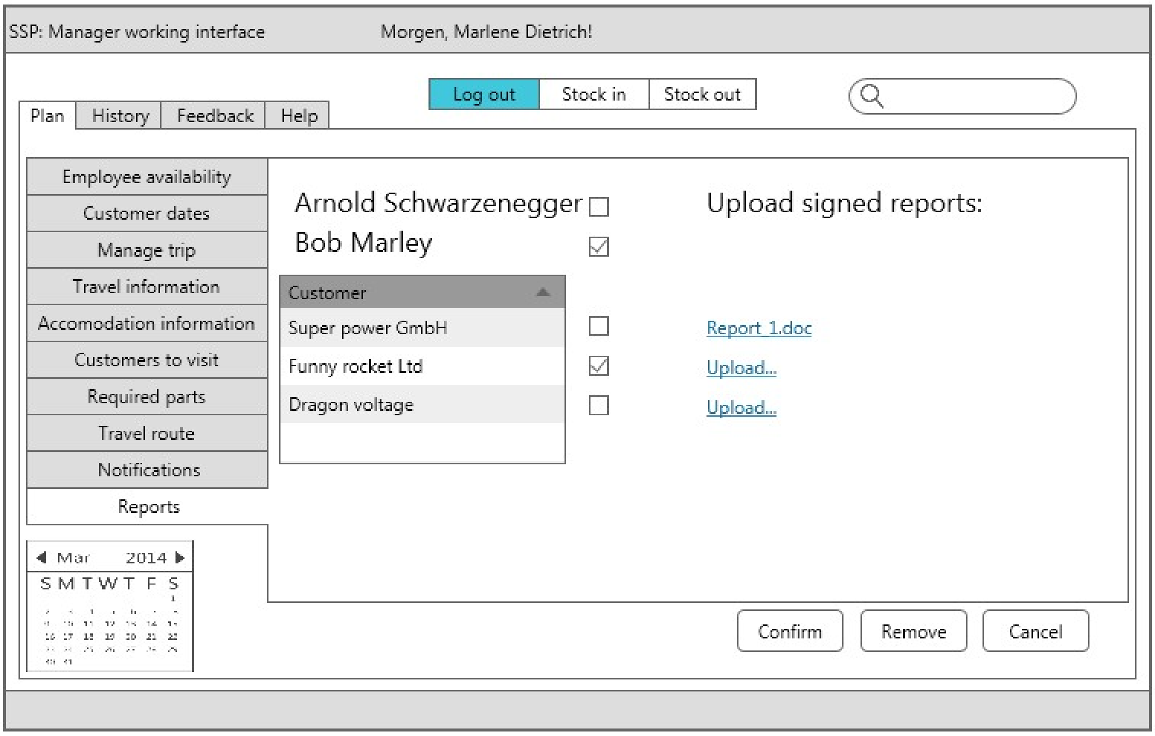
Spare parts

| Required parts(partnumbers) | Location on stock | Ordered | Delivery time |
|---|---|---|---|
| OCM board(PC98250) | Not available! | yes | 7 days |

Figure 12. Interface prototype „Notifications“ tab view.

Selecting „Notifications“ tab service manager can select first the service engineer making a tip on corresponding checkbox and see all notifications which belongs to selected engineer. Also there is possible to see the status of ordered parts and the delivery time information.

# Appendix 15 – Interface prototype „Reports“ tab view.



Arnold Schwarzenegger □
Bob Marley ☑

Upload signed reports:

| Customer | | |
|---|---|---|
| Super power GmbH | □ | Report 1.doc |
| Funny rocket Ltd | ☑ | Upload... |
| Dragon voltage | □ | Upload... |

Figure 13. Interface prototype „**Reports**“ tab view.

Selecting „**Reports**“ tab service manager can select first the service engineer making a tip on corresponding checkbox and see all performed service jobs by selected engneer. After engineer gives the signed by him and by the customer report to service manager there is a possibility to upload signed report.