Software Requirements Specification

for

iBISS Travel

Version 1.2

Prepared by Team 6

Members: Maximilian Meier, Hamoun Mojib, Frederic Lapatschek

06/06/2017

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Team 6	16/05/2017	Initial Draft	1.0
Team 6	19/05/2017	Final Version	1.1
Team 6	06/06/2017	Bug Fixes	1.2

1. Introduction

1.1 Purpose

The purpose of this document is to provide requirements for the iBISS Travel application and to provide information necessary to its development.

1.2 Intended Audience and Reading Suggestions

The intended audience for this document is Itestra, the project sponsor, the Android lab organizers, and the Android lab development group. Throughout the rest of this document the project will be broken up into sections for: Project Description, System Features, External Interface Requirements, and Non-Functional Requirements. For the formulation of requirements this documents adheres to RFC 2119 for the usage of the keywords “must” and “shall”. There is also a glossary of common terms found throughout the document attached to the document as an appendix.

1.3 Project Scope

This project’s goal is to develop an utility application for Österreichische Hagelversicherung’s insurance appraisers in coordination with Itestra, who retains the insurance company as a client. The application will assist the appraisers and salesmen with managing their work travels by automatically generating itineraries based on their appointments. It will also assist in handling expenses and the managing of appointments.

The software being used for development is the Android Studio provided by Google. A GitLab repository hosted by the LRZ will be used for source code and its integrated task management functionalities will be used as well. The backend server will be managed by Itestra and any required changes to the backend will be handled by Itestra as well.

2. Overall Description

2.1 Product Perspective

This application is being developed for the use by employees of Österreichische Hagelversicherung. The SV/VT (Sachverständige/Vertreter) departments will use this application to easily plan their trips to customers by different means of transportation. The SB (Sachbearbeiter) department will use this application to view and approve expense reports, specifically with a focus on employee reimbursements. The app will help to find the optimal route to each appointment. The app will allow the employees to interact directly with their appointment and expenses.

2.2 Product Features

This program will allow users to manage their work travel from their mobile phone. Any phone that supports the Android operating system version 4.4 or above will be able to install the applications and run the application from their phone. These tools will allow users to delegate the majority of their work travel management to their smartphone and thereby saving the users' time. The app will be responsible for scheduling appointments and saving automatically generated expenses of the employee.

2.3 Operating Environment

The software will run on the Android operating system version 4.4 (KitKat). All devices that support this version of the Android operating system will be able to run the software. The software is developed using the latest version of Android Studio. The intent of this software is to utilize the employees of Österreichische Hagelversicherung to find the optimal routes in terms of travel time, price, and distance for their working trips by train, car, or other public transportation. To fulfill this, the application will use different APIs to connect each functionality and use a provided backend system to save all information in a persistent data storage.

2.4 Design and Implementation Constraints

The design must be chosen with the employees in mind. The UI must be easy to understand and to use. The application must support the provided interface with existing Java Enterprise code. The functionality of the app is enhanced by using different sensors like GPS to provide better navigation for the employee.

2.5 User Documentation

A user documentation will be provided within the the app to explain the core functionality and describe how the user may make full use of its features. This documentation will be available as a pdf as well, which outlines the functionality and use cases for the employee. To help with further development the code will be documented as well and unit tests will be added to provide better code quality.

2.6 Assumptions and Dependencies

The system is dependent upon the provided backend system with the existing information of appointments and users. The system also uses existing authentication information.

3. Functional Requirements

F-1: Users shall be able to authenticate themselves with their existing credentials.
F-2: Users shall be able to logout of the application and invalidate their authentication.
F-3: The application shall load account, appointment, todo, and expense report data associated with the user from the backend upon authentication.
F-4: Users shall be able to view their associated account information.

F-5: Users shall be able to create/update/view/delete appointments.
F-6: Users shall be able to create/update/view/delete todos.
F-7: When creating, or updating an appointment or todo, the user shall be able to select from autocomplete suggestions for location information.
F-8: Users shall be able to manually start the todo and appointment synchronization process.
F-9: In the event of network loss, current data shall be locally saved until network connection and synchronization capabilities are restored.
F-10: Users shall be able to view an overview over their appointments and/or todos.
F-11: Users shall be able to amend todos by date and/or location information to make them appointments.
F-12: Users shall be able to delete date and/or location information from appointments to make them todos.
F-13: The application shall be able to generate and suggest multiple itinerary recommendations based on lowest costs and fastest routes for an appointment to the user.
F-14: The application shall be able to inquire about the user's home location.
F-15: The application shall use the user's home location as origin location for generating the itinerary suggestion for the first appointment of a day.
F-16: For a given appointment, the application shall use the preceding appointment's location for generating the itinerary suggestion.
F-17: Users shall be able to choose an itinerary suggestion for a given appointment.
F-18: Users shall be provided with booking links for itinerary segments if available.
F-19: Users shall be able to generate an expense report for a given past appointment.
F-20: Users shall be able to add items to or delete items from an expense report.
F-21: Users shall be able to edit items on an expense report.
F-22: Users shall be able to send an expense report to the backend system for approval.
F-23: Users shall be able to view expense reports that have not been accepted yet.
F-24: Users shall be able to view expense reports up to 7 days after they have been accepted.
F-25: Users' default currencies shall be determined by their chosen country during login.
F-26: All expenses shall be converted to the chosen currency at a fixed exchange rate.
F-27: The application shall notify the user about a day's appointments at the beginning of said day.
F-28: The application shall remind the user of the travel itinerary an appropriate amount of time before planned departure.
F-29: The application shall show users contact information of appointments and shall interact with default phone applications to call/mail/message appointments.

4. Constraints

C-1: The implementation must be in Java.
C-2: The implementation must run on Android 4.4 (KitKat) and above.
C-3: The implementation must run on API level 19 and above.
C-3: The implementation must interface with existing Java Enterprise code.
C-4: The implementation must interface with an existing backend system .
C-5: Dependencies' licenses must allow for commercial use or be licensed by the client.

5. External Interface Requirements

5.1 User Interface

UI-1: The user interface shall adhere to material design.
UI-2: The user interface shall accommodate for bigger end user fingers than might be usual.
UI-3: The user interface shall be structured in screens.
UI-4: The user interface shall consist of a login screen, account screen, appointment screen, todo screen, trip planning screen, expenses screen, help screen as well as any screen configurations pertaining to create/read/update/delete operations.
UI-5: Before logging in, the application will display the login screen which shall allow the user to select a country, currency or language.
UI-6: After logging in, the application will display the logged in view of the application which will be described by the following requirements.
UI-7: The user interface shall display an action bar with a hamburger menu icon as well as the name of the current screen.
UI-8: Upon clicking the hamburger menu icon a sliding menu bar will appear containing menu items that will navigate to the appointments, to-dos, trip planning, expenses, or help screen.
UI-9: The appointments, todos, and expenses screen shall display a (if applicable) chronological list of summarizations of a user's appointments, todos, and expenses in their default configuration as well as a floating action button that allows the user to add an appointment / todo / expense report.
UI-10: Every display of appointments, todos, expenses in a summarized format shall contain ui elements that allow for deletion and a detail view that contains detailed information and allows for editing.
UI-11: The trip planning screen shall allow users to book a train, map appointments, compare transportation costs,and view local public transportation options.
UI-12: The help screen shall provide users with an application user guide.
UI-13: Users shall be able to navigate through various screens seamlessly, without data loss.
UI-14: Users may log out of the system through the account screen.
UI-15: The expenses screen contains information regarding unreimbursed expenses and a short seven day history of closed expense tickets.
UI-16: At the beginning of every work day, the user shall receive a push notification relaying information about the first appointment of the day.
UI-17: All forms elements shall notify the user about what kind of input they expect.

5.2 Hardware Interfaces

This application is designed to be run on the Android system and therefore uses hardware commonly present on phones running the Android system. This application will make use of physical buttons found on the phone, such as the return button and the menu button.

5.3 Software Interfaces

This application will interface with a previously built backend system, associated Java Enterprise source code, various external API's such as the native Android services (Calendar, Phone, Mail, Messaging), various Google services, and transportation options. The application will interface with the native Calendar application to synchronize appointments and will interface with the native Phone, Mail, and Messaging application to communicate with appointments.

5.4 Communications Interfaces

This Android application will communicate with a multitude of services necessary for trip planning. As previously mentioned, this application will need to communicate with an external database and will use the Android Network and HTTPS in order to communicate. This application will make service requests to the backend server through Java Enterprise code. The application will need to be consistently synchronized with the external database via the backend system.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

PR-1: A stable and sufficient network must be available to synchronize with the external database.
PR-2: The backend system must be able to allow access to 300 concurrent users.

6.2 Security Requirements

SR-1: Data communication with the external database / backend system must be encrypted.
SR-2: Locally stored or cached data must be encrypted.

6.3 Software Quality Attributes

SQA-1: The entire application itself as well as subcomponents shall adhere to a greater software architecture as well as more local design patterns.
SQA-2: The source code shall adhere to the Google Java Style Guide as well as Android best practices.

7. Other Requirements

7.1 Data Requirements

DB-1: The information used in the application must be cached for offline use in an SQLite database
DB-2: The information saved on the Android device must be encrypted using the native Android system encryption.

Appendix A: Glossary

- *Appointment - A data object with information about the client, date, time and exact location data. It may also include an associated claim if applicable.*
- *Todo - An appointment with time and / or location data missing*
- *Expense Report - A list of expense items belonging to an appointment*
- *Expense Item - A data object with information about the type of transport, type of ticket as well as price*
- *Itinerary - A list of itinerary segments.*
- *Segment - A data object containing an origin location, destination location, departure time, arrival time as well as a type of transport and price/ticket information associated with it*
- *Claim - A ticket opened by an employee or customer with information such as customer information, damage reports, expense reports, etc.*
- *Costs - Company payed costs incurred by an employee in order to visit appointments*