Project Requirements

Project Name: MensaUniBe

Team: 6

Customer: Bledar Aga

Revision History

Version	Date	Revision Description
.01	2.10.13	Initial version of the SRS document
.02	9.10.13	Some general adjustments based on the customer feedback
.03	5.11.13	Actualization to the level of the app's v1 release
.04	4.12.13	General adjustments after all functionalities have been implemented
.05	11.12.13	Final version

Date: October 2, 2013

Introduction

Purpose

The purpose of this application (app) is to provide various informations about the different mensas of the university of Bern. These informations help the user to decide where to go for lunch, find any mensa on the campus or share his mensa experience with his friends by inviting them for lunch and/or rating menus.

Stakeholders

Users

Students and university employees that are looking for lunch and have an Android device. The users need to have an app which allows them to quickly find out everything they need to know about the mensas and the served menus.

Client

Our customer Bledar Aga represents the mensa company. Bledar needs a working app which is delivered within the time schedule.

Developers

The developers create a reliable app suited for everyday use whilst incorporating all the requirements communicated from the client.

System Overview

MensaUniBe Android App:

This is the interface for the users to the whole system. This app interacts with the Mensa-Webservice and our Social-Webservice.

Mensa-Webservice:

Webservice which provides all informations about the mensas and their daily or weekly menu plans. (https://github.com/lexruee/Mensa-Webservice)

Social-Webservice:

Handles the menu ratings, the invitations and the translation. Is separated from the external Mensa-Webservice. It has been realised with parse.com.

Bing Translation Service:

Microsoft's translation service is used to provide the automatic translations from the German menus to English.

Overall Description

Use Cases

Overview

0. Browse favorite mensas

Our average student or employee of the University of Bern has enough work to do besides looking for the best place to eat something as trivial as lunch. So to simplify this, he or she has the possibility to see all his or hers favorite mensas and the food they serve in just a glimpse at the moment they start the application.

1. Add mensas to favorites

In order to show all favorite mensas (see Use Case 0) the user actually needs to have a favorite mensa and also needs to add them to the system. So if the user wants to do this he can just press a button and the mensa is now a favorite. Pressing the same button again will remove the mensa from the list of favorites.

2. Browse all mensas

The user needs to get an overview of all mensas and their menus in order to select his favorites, find directions to any of them or just to look for decent food.

3. Browse all menus of the current day and rest of the week

The main meaning of the app is to have a daily or weekly overview of all the menus that are served in the mensas around the university of Berne. The app provides multiple ways to get such overviews. The users can either have a list of all menus of one mensa or they can see a list of all menus from the mensas they have chosen as favorites.

4. Find way to the mensas

The user wants to find the way to any mensa on the campus, so he is able to do so by just clicking the right button from the chosen mensa and gets guided. This feature includes that the closest mensa will be highlighted.

5. Invite friends for lunch

The user does not like to eat alone, so he decides to invite some friends for lunch. He starts the MUB-App and browses his options for lunch. As soon as he comes to a decision he clicks the “Invite Friends” button next to his menu or mensa of choice. In the following dialogue the user can choose who to invite by typing an email address from a friend and press the “Invite” button which then sends the invitation to his friend if an internet connection is available.

6. Reply to invitation

In the invitations screen the user can see whether he has been invited. Or he can look up his mail box because all invitations will be sent by mail, too. If there is an invitation the user can answer it and the other user will get the answer.

7. Check invitation status

The user sent an invitation to his friends some time ago, so he wonders if they come or not. In the “invited” list he sees all of his sent invitations. He clicks on the invitation of interest and sees the details of the invitation and also a list of persons he invited. Any person who accepted the invitation is marked with a green checkmark, friends who didn’t want to eat with the user are marked with a red cross. Anyone who didn’t answer yet just has a yellow question mark. If the user decides he wants to see more of his friends for lunch, he just presses the “Invite more” button and selects other friends which should receive the invitation from the friends list.

8. Rate a menu

Users want to share their experiences with others. That’s why they have the ability to rate the menus in the app. It’s just one click on the menu and a selection how many stars of five the menu should get. Anyone will be able to see the user’s rating, which supports the users in making the decision where to eat.

9. Setup notifications for menus matching certain criteria

Some users don’t want to check the menus every single day, because they might be a bit choosy of what they eat and they only find rarely something they like. So the app lets the user select criteria which he wants to be notified the day it’s served. This is achieved by selecting a set of criteria and a set of mensas in the apps settings.

10. Translate menus

There is a setting which allows the user to enable Bing translations of the daily menus. The menus are then translated into English. While the apps language is English the menus are only provided in German. Since the company running the mensas does not provide an English translation, that is why there is an automatical machine translation.

Diagram


MensaUniBe App
4. Find way to the mensas
include
2. Browse all Mensas
include
1. Add mensas to favorites
0. Browse favorite mensas
3. Browse all menus of the current day
include
8. Rate a Menu
5. Invite friends for lunch
6. Reply to invitation
7.Check invitation status
9. Setup notifications for menus matching certain criteria
10. Translate Menus
User
Google Maps
Mensa-Webservice
retrieve Data
Social-Webservice
Bing translation service

Detailed Use Cases

0. Browse favorite mensas

1. **Actors**
   User (primary)
   Mensa-Webservice (secondary)

2. **Description**
   The average student of the university of Bern wants to see his favorite mensas and is able to see their menus in just a glimpse.

3. **Trigger**
   The user starts the application by pressing the application's Icon.
   The application starts and automatically shows all the mensas' menus which are previously marked as favorite mensas.
   If there are no mensas marked as favorites, a text will appear that describes how to add favorite mensas.

4. **Preconditions**
   1. For optimal use favorite mensas must have been added.
   2. If the app is started for the first time an internet connection is needed.

5. **Postconditions**
   1. Student is now informed about the food served in his favorite mensas.

6. **Main Scenario**
   1. App retrieves mensa and menu information from the Mensa-Webservice if possible
   2. App shows the favorite mensas' menus they serve
   3. User has the information he needs

7. **Alternative Scenarios**
   4a. The user has no favorite mensas yet
      1. A description for adding favorites is displayed
      2. User follows the description and can add a mensa to his favorites by pressing the star button next to a mensa name

   4b. There's no connection to the Mensa-Webservice
      1. Message on the screen "Cannot download mensa data."
      2. The app displays the latest downloaded data.

8. **Notes**
   The following is valid in each use case but only described once in this use case:
   1. Starting the app.
   2. Each time the app is started it retrieves mensa and menu information from the Mensa-Webservice if there's a connection to the internet.

1. Add Mensa to Favorites

1. **Actors**
   User (primary)
   Mensa-Webservice (secondary)

2. **Description**
   If you are in the Mensa List or have selected the view ‘All mensas’ it is possible to add a mensa to your favorites.

3. **Trigger**
   The user either navigates to the Mensa List and then the application shows all the mensas which are previously marked as favorite mensas. Or the user selects ‘All mensas’ from the Spinner at the top in the Home screen.
   If there are no mensas marked as favorites, a description of how to add mensas will appear in the ‘Favorites’ view of the Home screen.

4. **Preconditions**
   1. User has a connection to the internet or has already downloaded the menus of the whole week
   2. The user is in the Home screen or the Mensa List

5. **Postconditions**
   1. From now on the mensa is shown as favorite mensa and when then app is started the mensa’s menus are shown

6. **Main Scenario**
   1. He or she searches for a mensa in the Mensa List
   2. The user presses the star button next to the chosen mensa’s name for marking as favorite
   3. User can now find this mensa’s menus under ‘Favorites’

7. **Alternative Scenarios**
   Adding a mensa directly from the Home screen
      1. No matter which view of the Home screen is displayed, the user can press the star button beneath any mensa to add this mensa to the favorites

8. **Notes**
   1. To remove the mensa the user just has to press the star button again. The color of the star will change between grey (not a favorite) and yellow (favorite).
   2. The user can also favorite more than one mensa.

2. Browse all mensas

1. **Actors**
   User (primary)
   Mensa-Webservice (secondary)

2. **Description**
   As a user I want to browse the mensas for their menus. That is, I want to switch between the mensas and get their menuplan for the whole week.

3. **Trigger**
   User is on the Home screen

4. **Preconditions**
   1. User has a connection to the internet or has already downloaded the menus of the whole week
   2. Home screen is displayed

5. **Postconditions**
   1. All menus of one mensa are listed on the screen.

6. **Main Scenario**
   1. User activates the spinner at the top and selects ‚Weekly plans'
   2. For one mensa all menus of the current day are displayed, the other menus are hidden behind the displayed days of the week
   3. By tipping on a day of the week its menus are displayed as scroll-down

7. **Alternative Scenarios**
   1.

8. **Notes**
   1. The user can switch between the different mensas by sliding his finger horizontally.
   2. A quick context sensitive search has been proposed.

3. Browse all menus of the current day and rest of the week

1. **Actors**
   User (primary)
   Mensa-Webservice (secondary)

2. **Description**
   a) As a user I want to see what menus are served at the current day.
   b) As a user I want to see what menus from the list of my favorite mensas are served at the current day.

3. **Trigger**
   User is on the Home screen.

4. **Preconditions**
   1. User has a connection to the internet or has already downloaded the menus of the whole week
   2. User has added at least one favorite mensa.
   3. User is on the Home screen

5. **Postconditions**
   1. All menus of the current day are listed on the screen
   2. All menus of the current day from the list of the favorite mensas are listed on the screen

6. **Main Scenario**
   a)
   1. A list of the current day's menus from only the favorite mensas is shown
   2. User selects ‚All mensas' from the action bar at the top
   3. All the menus served at the given day are sorted by mensa. If the user tips on one mensa its menus appear as a pull-down menu
   4. The User can switch between the days of the week

   b)
   1. If the user sees the list of ‚All mensas' or ‚Weekly plans' he has to choose ‚Favorites' from the spinner in the action bar
   2. User sees the list of today's menus (only those of his favourite mensas)
   3. The user can switch between the days of the week

7. Alternative Scenarios

1.

8. Notes

With a slider the user can also choose other days than the current day as long there are menus for these days (i.e. he will never see any menus for sunday as the mensas are only open during working days).

4. Find way to the mensas

1. Actors

User (primary)
Mensa-Webservice (secondary)
Google-Maps (secondary)
GPS-Module (secondary

2. Description

a) As a user I want to know where the mensas are.
b) As a user I want to get directions to the different mensas.

3. Trigger

a) User navigates to the “Navigation”-Tab
b) User clicks on the “Get directions” button next to a specific mensa

4. Preconditions

1. User has a connection to the internet
2. Connection to google maps is possible
3. GPS is turned on

5. Postconditions

a. All mensas are displayed in a map
b. The directions to the selected mensa is displayed on a map

6. Main Scenario

1. Trigger a) or b) applies:
2a. All available mensas are displayed on a map
3a. Clicking on the mensa opens an overview trigger
2b. The directions between the phone’s location and the selected mensas are displayed on a map

7. Alternative Scenarios

1. If there's no connection to the internet:
   Message on the screen "No connection to the internet."
2. If no GPS signal:
   Message on the screen "No GPS signal."
3. Google Maps not accessible:
   Message on the screen "No connection to mapping service."

8. Notes

1. Google maps will probably give good results. No alternative needed.

5. Invite friends for lunch

1. **Actors**
   User (primary)
   Social-Webservice (secondary)

2. **Description**
   As a user I want to invite some friends for lunch in a mensa.

3. **Trigger**
   The user decides to invite some friends

4. **Preconditions**
   1. User has a connection to the internet
   2. The user has already added some other users as friends

5. **Postconditions**
   1. The invitation is sent to the friends
   2. The invitation gets listed as a sent invitation in the “invites” list

6. **Main Scenario**
   1. User clicks on the “invite friends” button next to a menu
   2. A form to specify the event details shows. Mensa name, date and time are automatically filled from the informations of the chosen menu
   3. User can change those attributes, then choose some of his friends to be invited and write a personal message.
   4. The Social-Webservice saves the invitation for the invited users so that they can download it as soon as they are online

7. **Alternative Scenarios**
   Social-Webservice not available: App gives error message “Connection to Social-Webservice not possible”

8. **Notes**
   1. The alternative would have been Facebook.

6. Reply to invitation

1. **Actors**
   User (primary)
   Social-Webservice (secondary)

2. **Description**
   As a user I want to answer my invitations.

3. **Trigger**
   The user navigates to “Invitations”

4. **Preconditions**
   1. User has a connection to the internet
   2. The user has at least one invitation

5. **Postconditions**
   1. The answer to the invitation is sent to his friend
   2. The invitation stays as accepted in the list until the invitation time is over or it

disappears if the user has declined it

6. **Main Scenario**
   1. User clicks on the invitation he wants to answer
   2. A form to choose either accept or not opens
   3. The user makes his decision
   4. The answer is sent to the the Social-Webservice from which the other user will download the answer as soon as he is online

7. **Alternative Scenarios**
   Social-Webservice not available: App gives error message “Connection to Social-Webservice not possible”

7. Check invitation status

1. **Actors**
   User (primary)
   Social-Webservice (secondary)

2. **Description**
   As a user I want to see the replies to my invitation(s).

3. **Trigger**
   The user navigates to the “invites” tab in the "Invitations”

4. **Preconditions**
   1. User has a connection to the internet
   2. User has sent at least one invitation which has not passed yet

5. **Postconditions**
   1. User knows who accepted, declined or ignored the invitation

6. **Main Scenario**
   1. User clicks on an invitation he sent
   2. App retrieves replies from the Social-Webservice
   3. App opens a dialog which shows the details of the invitation and the replies

7. **Alternative Scenarios**
   2a. App cannot connect to Social-Webservice
   App shows an error: “Cannot retrieve replies at the moment, please try again later”

8. Rate a menu

1. **Actors**
   User (primary)
   Social-Webservice (secondary)

2. **Description**
   As a user I want to see the ratings of a menu and be able to rate it.

3. **Trigger**
   User opens the app with the intention to rate a menu

4. **Preconditions**
   1. User has a connection to the internet

5. **Postconditions**
   1. The rating has been sent to the Social-Webservice.
   2. The MensaUniBe app has updated the ratings so that the user is sure his rating has been uploaded

6. **Main Scenario**
   2. User navigates to the menu he wants to rate
   3. User taps on the Rate-Stars to define how many stars he wants to give
   5. App sends the rating to the Social-Webservice
   6. App downloads the rating-content from the Social-Webservice
   7. The ratings from the displayed menu are being updated and the user can see his new rating

7. **Alternative Scenarios**
   If there's no connection to the internet:
   1. Rating screen does not open
   2. Message on the screen "Connection to the internet required."

8. **Notes**
   1. If there's no connection to the internet, the Rate-Button could also be greyed out so that it couldn't be activated.

9. Setup notifications for menus matching certain criteria

1. **Actors**
   User (primary)

2. **Description**
   The user has the possibility to select criteria he wants to be notified of if a menu that matches is served that day. He can choose to be notified of all the mensas or his favorite mensas.

3. **Trigger**
   User turns on the “notify me” swich in the app’s settings

4. **Preconditions**
   If the user wants to be notified of his favorite mensas, favorites must have been chosen before. If he hasn’t chosen favorites before a message will be displayed.

5. **Postconditions**
   From this on the application sends push messages via android notification when a menu in the selected mensa set matches a criteria

6. **Main Scenario**
   1. User selects a set of mensas (all mensas or favorite mensas)
   2. User presses the button to edit the criterias
   3. A dialogue box appears
   4. In the text field the user enters a criteria (this should be the name of some food) and taps on the add button
   5. The criteria is added to a list at the bottom where it can be removed again

6. Now the user will get a notification message on his phone as soon as a menu description contains one of the user's criterias

7. Notes

1. When the notification is clicked, the user sees a view of all the matching menus and where they are served

10. Translate menus

1. Actors

User (primary)
Bing translation service (secondary)

2. Description

The user has the possibility to enable a setting so that all menus are translated to English by Bing.

3. Trigger

User is in the settings menu

4. Preconditions

The menus have been downloaded before they can be translated

5. Postconditions

All menus are displayed in their translated form

6. Main Scenario

1. User enables "Translate menus to English" in the app's settings
2. The app starts a task in background which translates all menus
3. As soon as all all menus are translated they are displayed translated

Specific requirements

Functional requirements

- Retrieve and display menu data from the Mensa-Webservice.
- Cache menu and mensa data to allow the app to be used without internet connection (until it is quitted).
- Get location of a specific mensa and show this mensa on the map (using a third party map API, i.e. google maps).
- Calculate directions and distance to a mensa.
- Manage the list of favorite mensas.
- Manage menu criteria and notify user if a menu matches the user's criterias.
- Translate mensa menus into English.
- Send message to other users (invitation or reply to invitation) using the Social-Webservice.
- Social-Webservice needs to distribute user messages to the addressees.
- Retrieve messages from the Social-Webservice.
- Send a rating to the Social-Webservice and receive ratings from the Social-Webservice. The ratings need to be associated with a menu's string representation to reuse ratings if a menu is served again.
- Manage a list of friends who also use the MUB-App.

Non-Functional requirements

- The Mensa-Webservice and Social-Webservice should be available. If not, the app can only display already downloaded data and not manage ratings, invitations and replies.
- The app should be self-explanatory.
- It should be more comfortable to view the mensa menus via the MUB-App than the mensa homepage. If not, there is no use for the app.
- The project should be finished in time.