

## D.2.1 Functional Specifications for the Prototype & Integrated Mobile Technologies



### Gamified Collaborative Platform for the Promotion of Sustainable Care and Independent Assisted Living

Grant Agreement No.	AAL-2014-79
Project acronym	MyMate
Start date of project (Dur.)	1 november, 2015 (24 months)
Document due date :	24 February, 2016 – Month 14
Leader of this report	Brainstorm
Deliverable No.	D.2.1
Deliverable Name	Functional Specifications for the Prototype & Integrated Mobile Technologies
Dissemination Level	Public
Contact Person	Francisco Ibáñez
Abstract	The purpose of this document is to describe the requirements for PC and mobile devices, the game needs, the story board, the user database and the monitoring and reporting processes.
Version	1.0



## EXECUTIVE SUMMARY

One of the most critical requirements that shapes MyMate platform is to make the system easy to use for older persons, usually with no previous experience with technological devices. So, the design of the architecture, workflow and user interface must be simple and clear, but complete enough to allow all the interactions and services required for the project.

The objective of this document is to detail the structure of MyMate platform, the software modules and the hardware architecture. The document is structured in several parts, from the design of roles to the workflow of the different services developed for MyMate.

<b>Executive summary .....</b>	<b>2</b>
<b>1. Introduction .....</b>	<b>4</b>
1.1. Document description .....	4
<b>2. User profilings .....</b>	<b>5</b>
2.1. Primary users .....	5
2.2. Secondary users .....	5
2.3. Tertiary users .....	5
<b>3. Functional specifications .....</b>	<b>7</b>
3.1. Desktop application .....	7
3.2. Mobile application .....	8
<b>4. Hardware and software requirements .....</b>	<b>9</b>
4.1. Personal computer .....	9
4.2. Mobile device .....	9
4.3. Server .....	10
<b>5. Application architecture .....</b>	<b>11</b>
5.1. Server architecture .....	11
5.2. Mobile application architecture .....	13
<b>6. Database architecture .....</b>	<b>17</b>
6.1. Users, availability and messages .....	17
6.2. Missions and reports .....	20
6.3. Rewards system .....	22
<b>7. Game mechanics .....</b>	<b>24</b>
7.1. Missions .....	24
7.2. Points .....	27
7.3. Levels .....	27
7.4. Rewards .....	27
7.5. Rankings .....	28
7.6. Assessment and reporting .....	28
<b>8. Storyboard .....</b>	<b>30</b>
8.1. Login .....	30
8.2. Mission income .....	31
8.3. Mission overview .....	32
8.4. Mission accomplishment .....	33
8.5. Mission report .....	33
8.6. Profile management .....	35
8.7. Chat .....	36
8.8. Rewards manager .....	36

## 1. INTRODUCTION

### 1.1. Document description

This document describes the functional specifications of MyMate system.

The structure of the document is organized in several chapters:

The User profiling chapter describes the different user roles that will be provided on MyMate application, explaining their capabilities and functions that each of them can make use of.

The Functional specifications chapter describes the functionalities that will be developed for the two user platforms, the Computer and the Tablet. The two platforms will have different capabilities and levels of management depending on the role of the user.

Next chapter, Hardware and software requirements, will describe the software and hardware requirements for the devices used for MyMate.

The Application architecture chapter explains the full software architecture of the applications, describing all the modules and functionalities of the platforms. This section is divided in two parts addressing the two main software applications that are going to be implemented: the server and the clients developed for the tablets.

The Database architecture chapter describes the database modules, and is divided in three sub-sections. This chapter explains all the tables and fields of the database.

The Game mechanics section describes the processes of the application and the corresponding workflows for the different user roles.

Finally, the Storyboard chapter provides snapshots (mock-ups) of the tablet application and describes each of them.

## 2. USER PROFILINGS

There are three different user levels for MyMate application. Each one of them will use a different application and have their own role.

### 2.1. Primary users

The primary users, also called 1<sup>st</sup> users, will be the people to take care of, usually persons staying at a care center or persons living in their homes that do not have enough capacity to perform all their daily activities independently.

These users will not have access to the system. For changing their personal data or their interests these people have to contact the caregiver of their care center that has been made responsible for this task.

### 2.2. Secondary users

The secondary users, also called 2<sup>nd</sup> users or volunteers, will be the people with no specific training on older people care, in charge of assisting and accompanying the primary users in their daily activities.

This voluntary work will be called missions, and will be planned by 3<sup>rd</sup> users. By performing these actions, the secondary users get points that can be exchanged for gifts.

To do this job, the secondary users will access the system using the mobile application installed on their tablet devices.

### 2.3. Tertiary users

The tertiary users, also called 3<sup>rd</sup> users or formal carers, will be the specialized people in the care center that are in charge of taking care of the 1<sup>st</sup> users, preparing the missions for the 2<sup>nd</sup> users to help them in their needs, controlling whether the missions are successfully completed and asking the primary users about their experience with the volunteers.

They will also be in charge of managing the reward program and ensuring that it is working properly.

The administration of all those features will be done using the desktop application, which will be installed on a computer available at the care center.

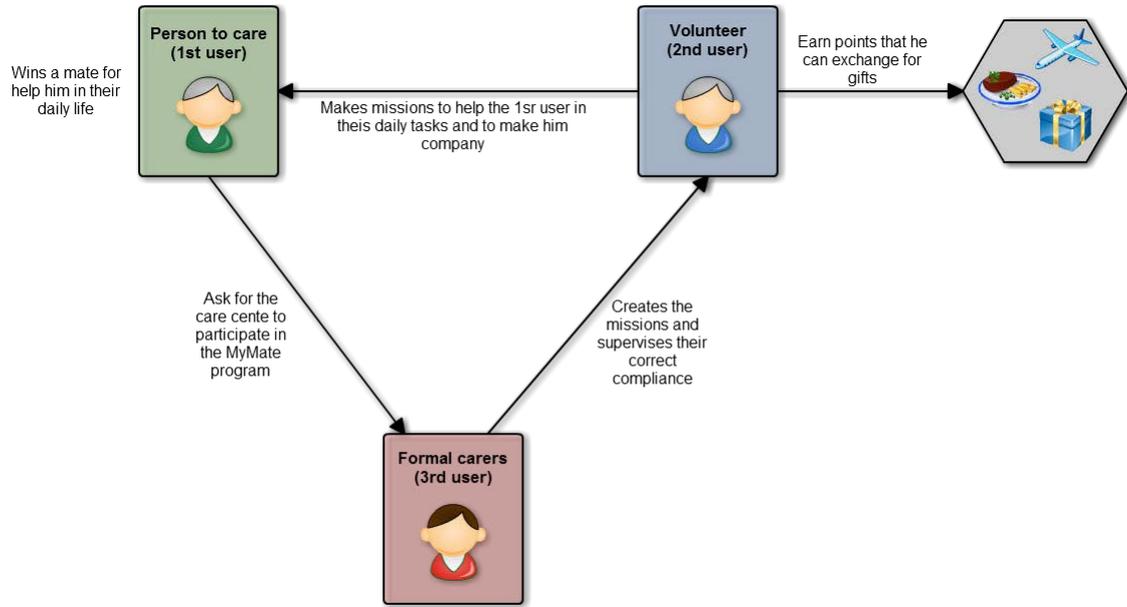


Figure 1- User profiles in MyMate

### 3. FUNCTIONAL SPECIFICATIONS

#### 3.1. Desktop application

The desktop computer will be used by the Tertiary users to manage the personal data and configuration of the primary and secondary users, supervise the missions and status of the activities, and control the rewards system. Those actions will be managed using a web-based application hosted on a central server. This chapter describes the functional specifications of the desktop application.

##### User management

The formal user is allowed to do CRUD (Create, Read, Update and Delete) operations to manage the first users and second users. The administrator of the care center will also have additional permissions to do CRUD operations to manage the formal carers.

The configuration of the users will have the following parameters:

- 1<sup>st</sup> users data
  - Personal data: Name, date of birth, contact details, address and picture.
  - User interests. This data will be collected using some questionnaires to get information about their hobbies and habits. This data will be used to do a matching with the volunteers.
- 2<sup>nd</sup> users data
  - Password
  - Maximum distance from their home address they can afford to go.
  - Calendar-based availability
- 3<sup>rd</sup> users data
  - Personal details

##### Missions management

The formal carers should be able to manage the different aspects and configuration of missions. The main functionalities to be configured are:

- Mission creation: Create new missions, select the different users that will be involved in the mission and configure related aspects:
  - Activities to do in the mission.
  - Location where the mission will take place.
  - Date and time of the mission.
  - Points assigned to the mission.
- Mission cancellation: The mission can be manually cancelled at any point in time.
- Mission assessment: Once the 2<sup>nd</sup> user has completed the mission and the report has been generated, the 3<sup>rd</sup> user responsible should verify that the mission has been completed successfully and assign the corresponding points.

## Rewards management

The 3<sup>rd</sup> users will be responsible for the management of the awards granted to the 2<sup>nd</sup> users. The different tasks that the 3<sup>rd</sup> users can perform are:

- Talk to local supporters (restaurants, shops, travel agencies, etc...)
- Enter the rewards in the system
- Allocate points to rewards
- Publish the catalogue
- Allocate rewards to 2<sup>a</sup> users when they have enough points to redeem points
- Check the points & rewards history of 2<sup>nd</sup> users

## 3.2. Mobile application

### Mission execution

The users should be able to access the following functions related to the missions:

- Manage missions:
  - See all the assigned missions
  - Accept the mission(s)
  - See the pending missions
  - Start a mission
  - Finish a mission
  - Generate a mission report

The workflow and details about the mission accomplishment is described in detail in section "Game mechanics – Missions".

### Profile management

The 2<sup>nd</sup> users will be able to access the following functions:

- Access their profile, which stores the following information:
  - Full name
  - Photography
  - Contact information (Address, Phone, email, ...)
  - Ranking in the local center
  - Ranking on the whole program
  - Total, week, and last month score
  - Current level and necessary points to upgrade to the next one
- View his availability:
  - Calendar of current month, highlighting the days the user is available
  - Details of a specific day, by clicking click on that day

In order to make changes in their personal data or in their availability, the 2<sup>nd</sup> users should call the care center.

## 4. HARDWARE AND SOFTWARE REQUIREMENTS

### 4.1. Personal computer

As the desktop application will be opened in a web browser, the desktop computer requirements are basic enough. The application can be run on most of the current commercial computers. The minimum requirements are:

- Screen size 19" and 1440 x 900 pixels resolution
- 1 GB of dedicated RAM
- 1 GB of available disk space
- At least following versions of computer operating systems:
  - Windows 7
  - MacOS 10.8
  - Linux kernel 2.6
- Internet connection with a minimum bandwidth of 1 Mbps.
- At least following versions of web browsers:
  - Google Chrome v56
  - Mozilla Firefox v51

### 4.2. Mobile device

Regarding the mobile device, used by 2<sup>nd</sup> users to view their profiles and manage the missions and rewards, the recommendation is to use a tablet and not a mobile phone. The reason lies on the larger screen and higher resolution. The minimum requirements for this device are:

- Screen Size 10.1" and 1920 x 1200 pixels resolution
- Minimum RAM 1G, although it is recommended to have 2G. (The app needs 200 MB of RAM)
- Internal memory 16 GB (The app needs 100 Mb of space)
- GPRS access (SIM card). The device should be able to access at least one of the following networks: Infra 2G GSM, 3G WCDMA, 4G LTE FDD, 4G LTE TDD (Preferred networks are 3G and/or 4G)
- GPS and/or Glonass location hardware. The location information is required to check that the user is correctly performing the missions.

For testing activities, the selected device is the Samsung Galaxy Tab A 10, which meets the specified minimum requirements.



Figure 2 - Samsung Galaxy Tab A 10, device selected for testing the application.

#### 4.3. Server

The server will manage all the information stored on the database and will provide the client applications access to the platform services.

The following requirements are based on an estimated maximum traffic load of 100 requests per second and a database containing 2000 users, considering the possible three type of users at every center:

- Internet connection of symmetric 2 Mbps
- Single core x86\_64 processor (1.8 GHz)
- 2 GB of RAM
- 100 GB of Disk Drive
- Linux, Windows or MacOS Operating Systems supporting:
  - Nginx server 0.8.40 or later
  - Python runtime 2.7, 3.4 or 3.5
  - One of these database management system:
    - PostgreSQL 9.2 or higher
    - MySQL 5.5 or higher

## 5. APPLICATION ARCHITECTURE

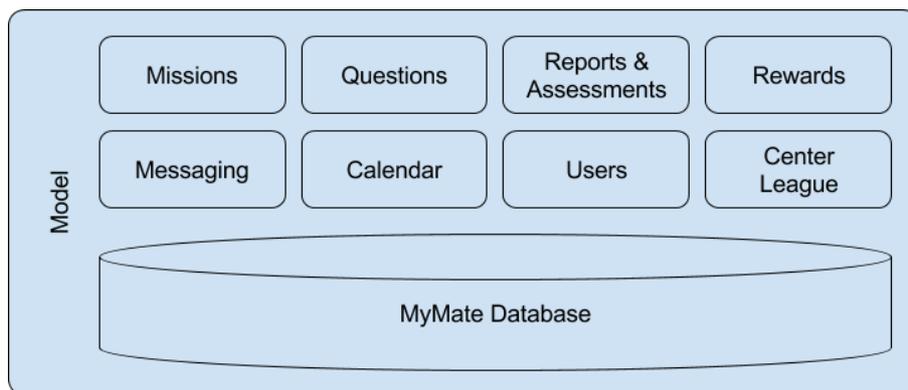
### 5.1. Server architecture

The web application uses a modular model to allow reusing the modules in different parts of the system.

#### Data Model

The data model makes transparent to the application the access to the database. This model defines the calls and procedures to run CRUD operations on following data:

- Missions
- Questions
- Reports & assessments
- Rewards
- Messaging
- Calendars
- Users
- Rankings for a specific center



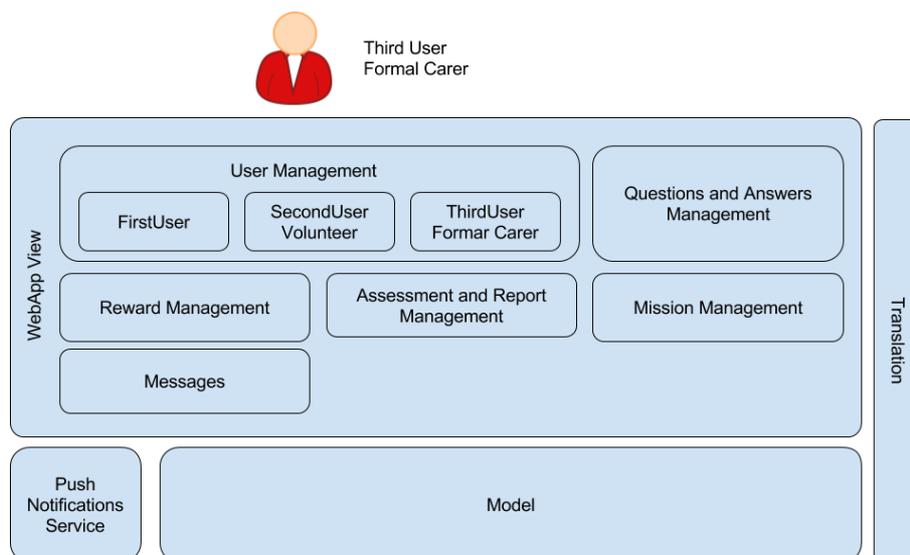
This model will be used by both the graphical interface of the web administration application and the API REST of the Android application.

#### WebApp View Dashboard

The web application is designed for the care center users, so that they can interact with the system. The components of the web application are:

- User management. There will be three GUIs for the administration of the three types of users:
  - 1<sup>st</sup> users: GUI for the administration the 1<sup>st</sup> users' data. The inputs requested by this GUI are the personal details and interests, which will be used in other parts of the interface to create missions.

- 2<sup>nd</sup> users: GUI for the administration of the 2<sup>nd</sup> users' data. The inputs requested by this GUI are the interests, the availability calendar and maximum allowed distance from the home location.
- 3<sup>rd</sup> users: GUI for the administration of the 3<sup>rd</sup> users' data. This GUI is only available for administration users.
- Questions and answers: GUI to manage the questions made to the volunteers and their answers, upon a completion of a mission.
- Reward management: GUI to perform CRUD operations for reward items.
- Assessment and report management: module to display the reports generated by the volunteers and evaluate the completed missions so that the corresponding points can be granted.
- Mission management: module to create and configure missions, change their status, etc.
- Messages: Module to display the body and destination user of messages sent and received by the user. This module can also create new chats with other users.

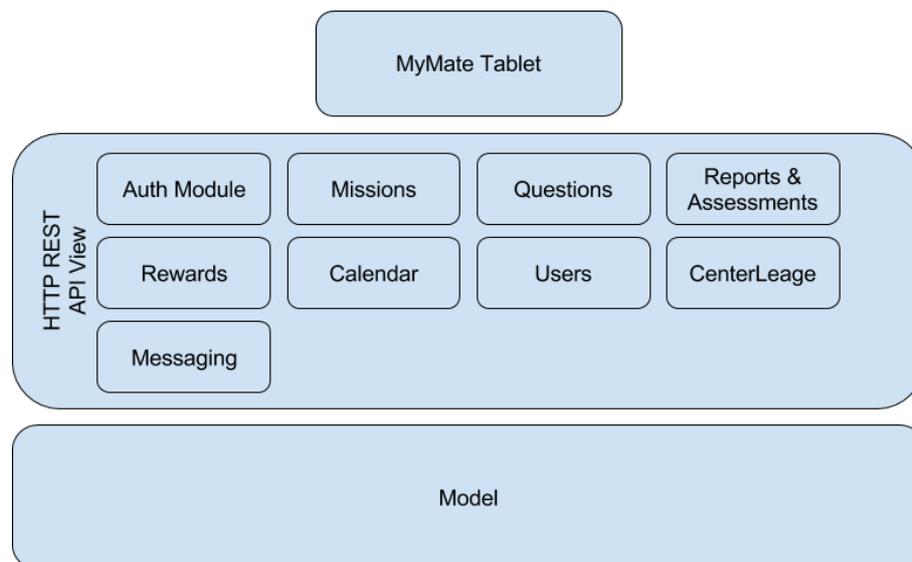


### HTTP Rest Server

The MyMate application installed on the tablet uses the http API to display the data of the missions, authenticate the users and run the operations required by the volunteers. This API offers the following functionality:

- Authentication: validation of user identity.
- Missions: interface to access the mission's data and create and modify its status.
- Questions: provision of questions for a specific mission.
- Reports and assessments: generation and sending of mission reports.
- Rewards: Visualization of the rewards that can be afforded by the user and the purchase feature.

- Calendar: retrieval of user availability and list of missions assigned to a specific date.
- Users: retrieval of relevant user information.
- Ranking: retrieval of general user ranking and position for a specific user.
- Messaging: visualization of the messages sent and received by the user and the functionality to send new ones.



## 5.2. Mobile application architecture

The mobile application will make use of the services provided by the application server. The application architecture has been designed taking into account that almost any operation initiated by the user inside the application will require an enquiry to the application server.

To meet this requirement, the design of the application needs to follow a modular approach. The planned modules will be classified in the following two main categories:

- **Vertical capability:** Any call to a module with vertical capability will start a process and a call to a module of the immediately inferior level. There will be two groups of modules with this capability in the mobile application:
  - **Graphical User Interface (GUI):** The GUI modules will implement the interface with the user and will be responsible for receiving the user queries, e.g. operations associated to buttons.
  - **Remote Procedural Call (RPC):** The RPC modules will implement the functionality to communicate the server with the network. Most of these modules perform intermediary functions between the graphical interface and the server.

- **Transversal capability:** The functionalities provided by modules with transversal capability are used by any other kind of module inside the application.

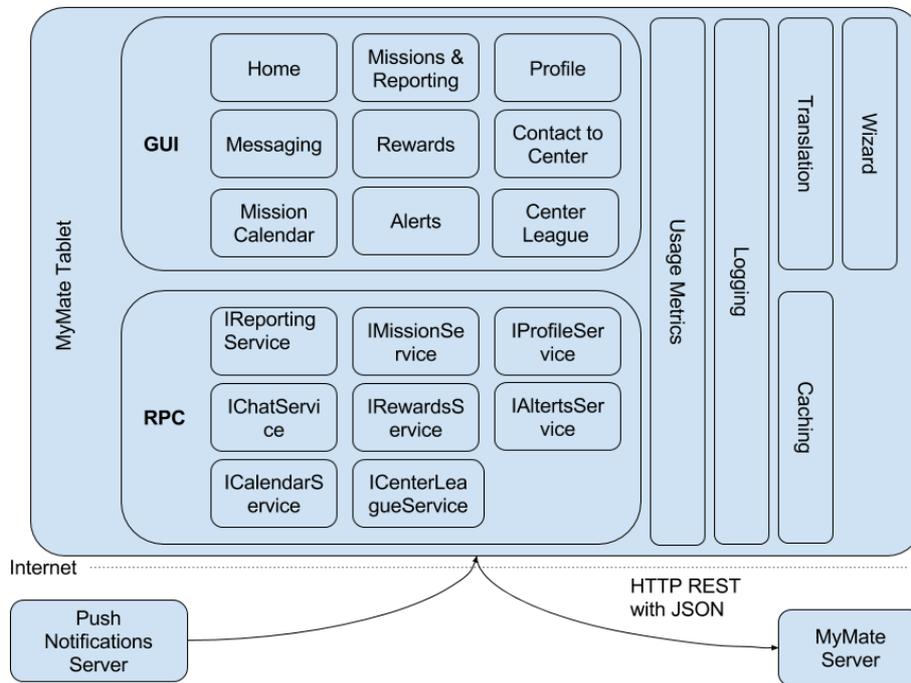
The following list describes all the modules of the application:

- **Vertical RPC modules:**
  - **IReportingService:** This module provides an API for the management of the mission reports. The users of the API, e.g. a GUI module, will use this API to retrieve the relevant mission forms and to send the inputs from the user, e.g. the form of the mission filled in, to the server.
  - **IMissionService:** This module provides an API to manage the missions. This API will be used by GUI modules to perform all the allowed operations, e.g. Get the list of active missions, modify the state of a report, etc.
  - **IProfileService:** This module provides the GUI modules access to the user data, e.g. performance score.
  - **IChatService:** The GUI modules will use the services provided by this module to send, receive and open messages.
  - **IRewardsService:** This module allows the GUI modules to access the rewards system and provides the necessary calls to purchase the rewards.
  - **IAlertsService:** This service provides Access to push notifications.
  - **ICalendarService:** This module provides the API to get the user availability information and the missions assigned to following days.
  - **ICenterLeagueService:** This module provides the API to get the ranking and query the position in the ranking for a specific user.
- **GUI modules:** They shows the information to the user and provide the interaction interface:
  - **Home:** This module provides the user home screen. It makes use of following RPC modules:
    - IMissionService
    - IProfileService
    - ICaldenarService
    - IAlertService
  - **Missions:** This module displays the missions' data and reports. It makes use of following RPC modules:
    - IMissionService
    - ICalendarService
    - IReportService

- **Profile:** This module displays the user data, e.g. the user level, and the ranking information. It makes use of following RPC modules:
  - IProfileService
  - ICenterLeagueService
- **Chat:** This module displays the messages that have been sent or received by the user, and provides the mechanisms to send new messages. It makes use of following RPC modules:
  - IChatService
  - IAlertsService
- **Rewards:** This module displays the available rewards and the associated price in points and allows the user to make a purchase. It makes use of following RPC modules:
  - IRewardsService
  - IProfileService
- **ContactToCenter:** This module allows the user to contact the center staff. It makes use of following RPC modules:
  - IProfileService
  - IChatService
- **Transversal capability modules:**
  - **UsageMetrics:** This module will monitor how the users use the application and whether the application works properly.
  - **Logging:** This module will log the events generated by the application
  - **Translation:** This module will display the graphical interface in the language configured by the user.
  - **Wizard:** This module will help the user in the use of the application. This module can be enabled/disabled by the user.
  - **Caching:** This module will be a kind of buffer used by the server to reduce the use of bandwidth.



First User  
Volunteer



## 6. DATABASE ARCHITECTURE

For an optimized management the database structure has been divided into three different sections. The different tables and fields that compose the database for each of those sections are detailed in this chapter.

### 6.1. Users, availability and messages

This part of the database will store all the information of the users and will implement the communication between them. The structure of the tables is as follows:

- **User:** The User table models the abstract users and contains the common fields of all users. The users can belong to one of the following types of users:
  - **FirstUser** (Elderly for Care)
  - **SecondUser** (Volunteer Elderly - Informal Carer)
  - **ThirdUser** (Formal Carers - Center's staff)

The fields of the table are:

- **Id:** Unique identifier for the user.
  - **Center:** Reference key of the center the user is related to.
  - **Login:** Alias of the user.
  - **Password:** encrypted password.
  - **Personal data fields:** **Name, second name, surname, birth date, address** (reference to the user's address stored on the Address table), **phone number** and **e-mail**.
- **Address:** The Address table stores the users' complete addresses. Apart from the unique identifier, the following fields will be provided: **Description, street** and **number, ZIP code, city, state, country, latitude** and **longitude**, for matching with the GPS coordinates.
  - **FirstUser:** This table stores the list of MyMate's 1<sup>st</sup> users. This table does not have any other information as these users do not have access to the database.
  - **SecondUser:** This table stores the data of the 2<sup>nd</sup> users of MyMate. In MyMate's database, SecondUser will be a specialization of type User. This table also contains the following fields:
    - **TotalPoints:** Points obtained by the user throughout the missions.
    - **PointsAvailableToSpend:** Available points that the user can use to exchange for gifts.
    - **Medal:** Which medal the user has and whether or not he has any. Foreign key to the Medal table
    - **MobilePushID:** Identifier of the user's tablet so that they can be sent push notifications.
    - **DeviceType:** Field that stores the device type used by the user. By default it will be Android but it is saved for future uses.

- **ThirdUser:** This table stores MyMate’s third-party users. In MyMate’s database, ThirdUser will be a specialization of type User. This table does not contain any additional fields.
- **Center:** This table stores the care centers that are registered in the system. It has the following fields:
  - **Id:** Unique identifier of the center in MyMate’s system.
  - **Administrator:** Reference to the 3<sup>rd</sup> user in charge of the care center.
  - **Name:** Name of the care center.
  - **PathLog:** URL to the care center logotype image.
- **Messages:** This table stores the messages sent by the users of the application. The table has the following fields:
  - **Id:** Unique identifier of the message at a global level.
  - **Timestamp:** Date and time the message was sent.
  - **Sender:** Reference to the user that sent the message.
  - **Receiver:** Reference to the user that received the message.
  - **Message:** Content of the message.
- **Interests:** This table stores the interests of 1<sup>st</sup> and 2<sup>nd</sup> users. The idea of this table is to be able to do an event matching query at the time of selecting the missions. It has the following fields:
  - **Id:** Unique identifier.
  - **Description:** String describing the interest.
- **FirstUserInterests:** This table stores the interests of the FirstUser in the many-to-many way. This table has the following fields:
  - **FirstUser:** Foreign key of the FirstUser table that identifies the user.
  - **Interest:** Foreign key of the Interest table that identifies interest.
- **SecondUserInterests:** This table stores the interests of SecondUser in the many-to-many way. The table has the following fields:
  - **SecondUser:** Foreign key of the SecondUser table that identifies the user.
  - **Interest:** Foreign key of the Interest table that identifies interest.
- **SocialAuthService:** This table stores the necessary session data to be able to login into a social account, like Facebook, to share information. The table has the following fields:
  - **Id:** Unique identifier.
  - **Name:** Service name.
  - **Token:** Service identification key.
  - **GrantTimeStamp:** Last validation date.
  - **SecondUser:** Foreign key of the SecondUser table to which it refers.

- WeekDaysMeta:** This table is used to implement a simple recurring day calendar system. It will only have 7 entries, one for each day of the week. For each day, the table will store the start date of the replay and the time in seconds corresponding to the repetition. In this way, from a single table the repetitions of all events can be managed. The table has the following fields:
  - Id:** Day identifier (From 1 to 7)
  - day\_name:** English day name string.
  - repeat\_start:** First event date.
  - repeat\_interval:** Repeating interval in seconds.
- SecondUserAvailability:** Stores the time availability data for a specific day for a particular user. It has the following fields:
  - Id:** Interval identifier
  - SecondUser:** Foreign key of the SecondUser table that identifies the user.
  - WeekDay:** Foreign key of the WeekDaysMeta table that identifies the day and the repetition.
  - Start:** Start time of the interval.
  - End:** End time of the interval.

The result architecture of the database for the Users, availability and messages section is shown on Figure 3.

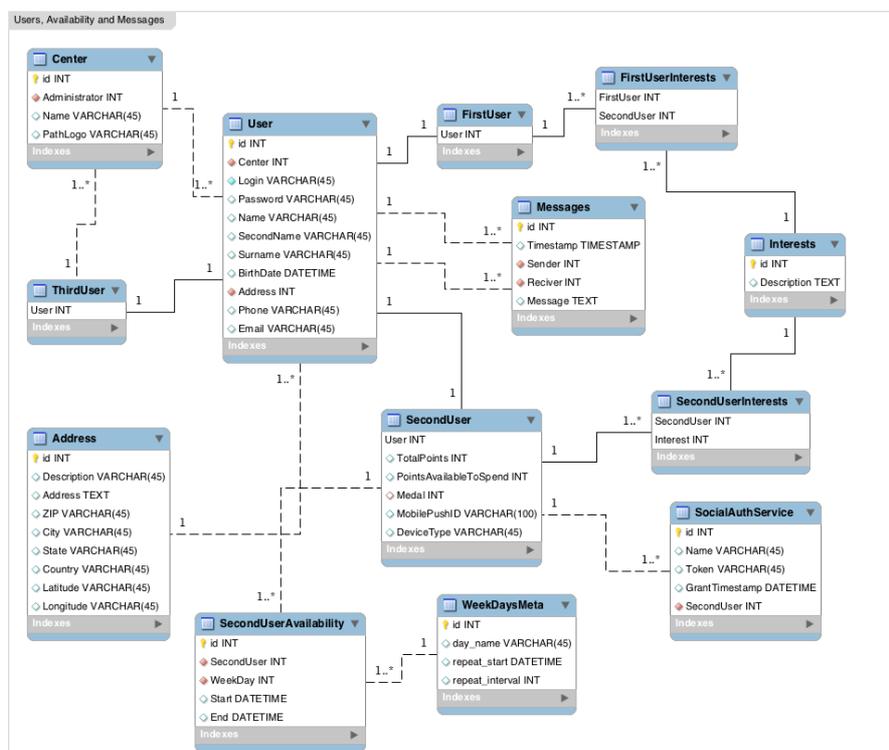


Figure 3 - First part database architecture.

## 6.2. Missions and reports

This part of the database will store all the information about the missions configuration, mission progress and reporting. It is structured in nine tables.

- **State:** It stores the possible states for a mission throughout its "life cycle". The field list is:
  - **Id:** Unique identifier.
  - **Color:** Each state will be defined by a color encoded in hexadecimal.
  - **Name:** Stat name in English.
  - **Description:** State description in English.
  - Also the table will contain fields to store the **name** and the **description** in Spanish, Romanian and Dutch.
- **RejectMissionOptions:** This table stores a list of possible reasons for rejecting a mission, so that the mobile application can display a drop-down list of allowed options. The list of fields is:
  - **Id:** Unique identifier.
  - **Center:** Foreign key referencing the Center table.
  - **Reason:** String describing the reason.
- **MissionStates:** It stores the current state for any mission. The fields are the followings:
  - **Id:** Unique identifier.
  - **User:** Foreign key to **User** table.
  - **Mission:** Foreign key to **Mission** table.
  - **State:** Foreign key to **State** table.
  - **When:** Timestamp about when the stat has changed.
  - **Comment:** String with the user comment.
  - **RejectMissionOption:** Foreign key to **RejectMissionOption**.
- **Mission:** In this entity, all the data of a mission are stored. The field relationship is as follows:
  - **Id:** Unique identifier.
  - **CreationData:** Timestamp about when the mission was created.
  - **StartDate:** Mission start time.
  - **Duration:** Estimated hours of mission duration.
  - **Activities:** The different tasks to do on the mission.
  - **Title:** Mission title.
  - **Description:** Mission description.
  - **Creator:** Reference to the 3<sup>rd</sup> user who creates the mission.
  - **SecondUser:** Reference to the 2<sup>nd</sup> user who the mission is assigned.
  - **FirstUser:** Reference to the 1<sup>st</sup> user who the mission is targeted.
  - **Address:** Location where the mission must begin.

- **CenterMissionEvaluation:** This table is used to store the evaluations that the center staff performs of the volunteer's performance and of the mission in general. It has the following fields.
  - **Mission:** Foreign key to the Mission table.
  - **FinalEvaluationOfSecondUser:** 2<sup>nd</sup> user affinity numeric assessment.
  - **Comment:** Free user comment.

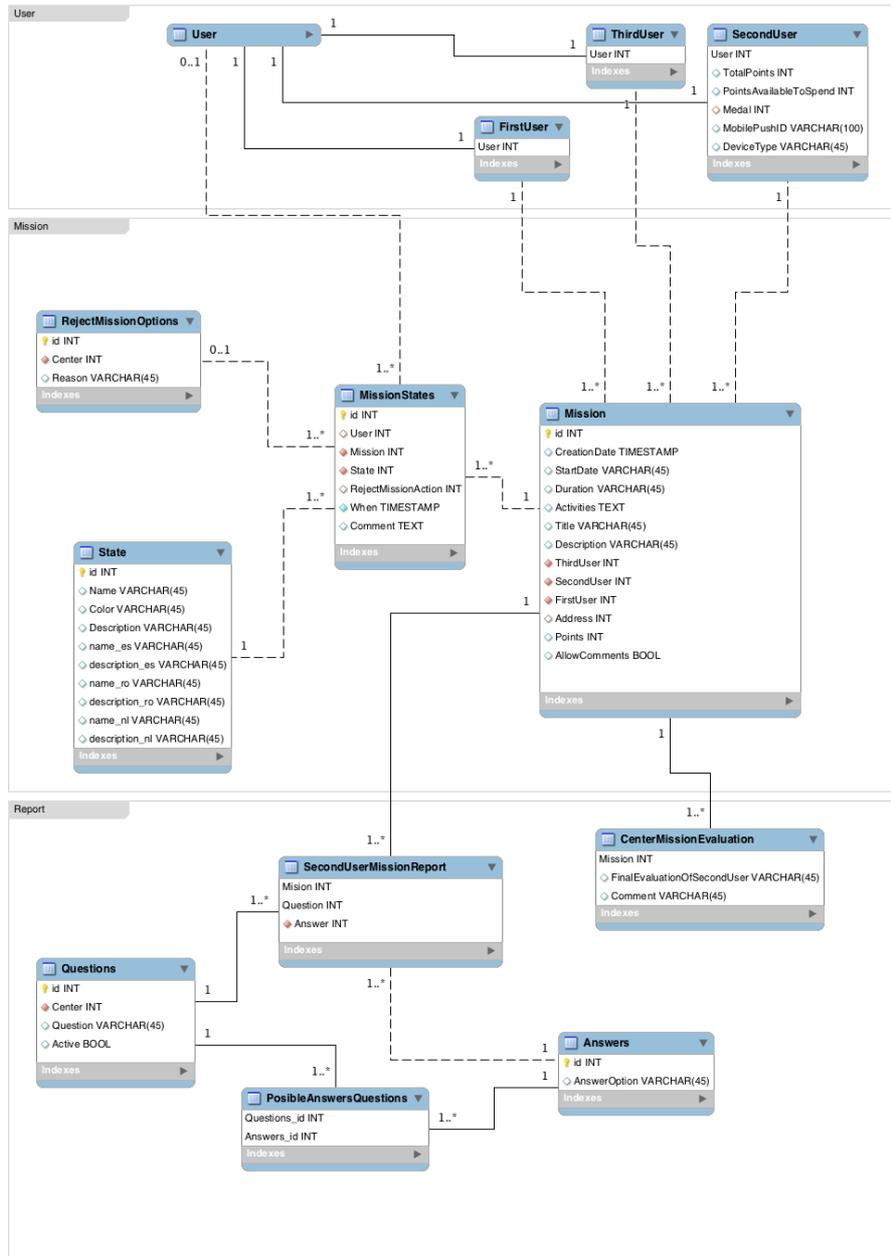


Figure 4 - Second part database architecture.

- **Question:** Table with the questions that will be asked during the missions. These questions have predefined answers and are the same for all the missions of a center. The field relationship is as follows:

- **Id**: Unique identifier.
- **Center**: Foreign key to the Center to which the question belongs.
- **Question**: Question string.
- **Active**: this field stores whether the question is active or not.
- **Answer**: This table will store all the possible answers to the questions. The corresponding fields are:
  - **Id**: Unique identifier.
  - **AnswerOption**: Answer string.
- **PossibleAnswerQuestion**: This table stores possible questions and answers pairs. The list of fields is as follows:
  - **Question**: Foreign key to Questions table.
  - **Answer**: Foreign key to Answers table.
- **SecondUserMissionReport**: This table stores user responses for the various missions The fields are the followings:
  - **Mission**: Foreign key to Mission table.
  - **Question**: Foreign key to Questions table.
  - **Answer**: Foreign key to Answer table.

### 6.3. Rewards system

This section of the database will store all the data relative to the scoring of the 2<sup>nd</sup> user, the level and the rewards management.

- **Medal**: Each of the medals that the user can obtain, depending on the points he has. It contains the next fields:
  - **Id**: Unique identifier.
  - **Name**: Medal name.
  - **Points**: Minimal scoring to get the medal.
  - **PicturePath**: Path to the image of the medal.
  - **Color**: Hexadecimal code for the color of the medal.
- **PointsAcquired**: List of points that each user has acquired in each mission.
  - **Id**: Unique identifier.
  - **User**: Foreign key to the SecondUser table.
  - **Mission**: Foreign key to the Mission table.
  - **When**: Timestamp about when the points were obtained.
  - **Points**: Obtained point quantity.
- **Reward**: It will contain the list of rewards than a user can exchange.
  - **Id**: Reward unique identifier.
  - **Center**: The center owner of the reward. Foreign key.
  - **Name**: Name of the reward.
  - **Where**: Location where the reward will be spent.

- **NumberOfUsesByUser**: Popularity of the reward.
- **RewardPriceByMedal**: This table assigns a different price to each reward based on the medal level:
  - **Medal**: Foreign key to Medal table.
  - **Reward**: Foreign key to Reward table.
  - **Price**: Reward cost in points.
- **RewardsBought**: This table will store the different rewards already acquired by the users.
  - **Id**: Unique identifier.
  - **SecondUser**: Foreign key to the user than acquires the reward.
  - **Reward**: Foreign key to the reward acquired.
  - **Price**: Point cost in the moment of the exchange.
  - **When**: Timestamp of the exchange moment.
- **RewardExchanged**: This table stores the reward purchase data.
  - **Reward**: Foreign key to the **RewardsBought** table.
  - **Code**: Identification code of the purchase.
  - **Exchanged**: Toggle about if the user has spent the reward.

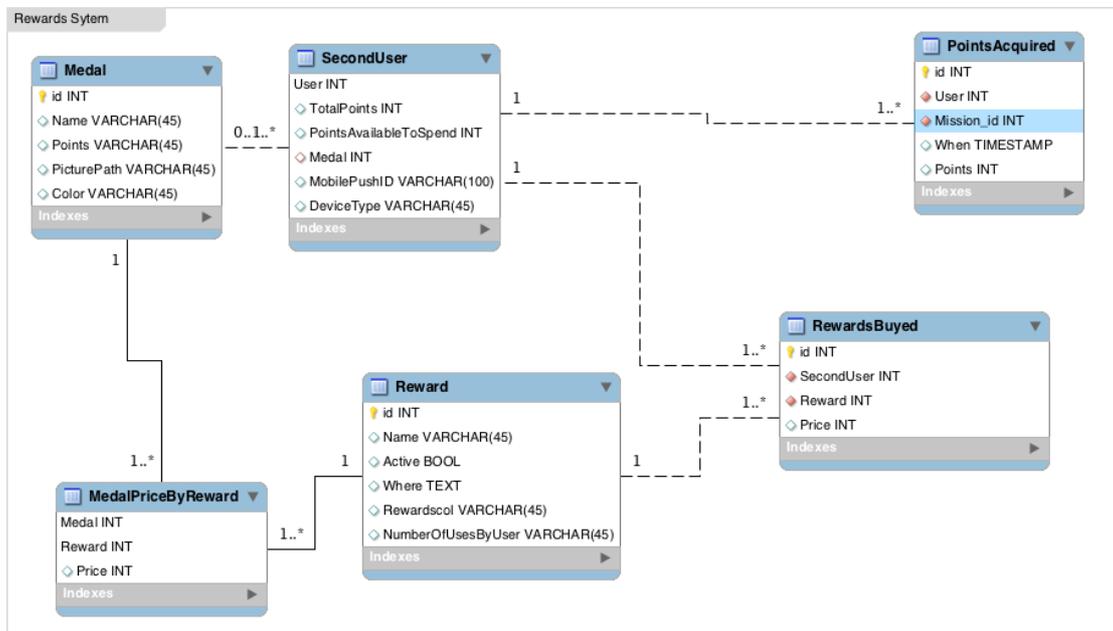


Figure 5 – Third part database architecture

## 7. GAME MECHANICS

### 7.1. Missions

#### 3<sup>rd</sup> users workflow

The 3<sup>rd</sup> user will be able to manage the missions. Thus, all the related actions they can do through the desktop application are:

#### **Create and assign missions.**

The 3<sup>rd</sup> users can create missions for the volunteers. The option to create a mission will open a wizard that will ask for the required information:

- First, the 3<sup>rd</sup> user should select the object of the mission, the FirstUser. The application automatically assigns the house of the 1<sup>st</sup> user as the MyMate destination. The wizard also asks for the date and time for performing the mission.
- Then, the 3<sup>rd</sup> user should configure the missions:
  - The missions have to be assigned to a 2<sup>nd</sup> user. Based on following parameters the application will provide a list of available 2<sup>nd</sup> users:
    - The 2<sup>nd</sup> user has no assigned mission at that time
    - The 2<sup>nd</sup> user is available for that time of the day
    - The distance between the 2<sup>nd</sup> user home and the MyMate destination is within the max allowed distance boundary.
  - Once the volunteer is selected, the list of matching interests and the level of achieved affinity will be displayed. This information will be helpful for configuring the mission. The parameters to be configured are:
    - String field describing the Mission title
    - String field describing the Activities to be performed
    - Numerical field to introduce the points to be assigned to the mission.
  - The last step is to confirm the mission creation operation. The system will create the mission and will send a notification to the volunteer.

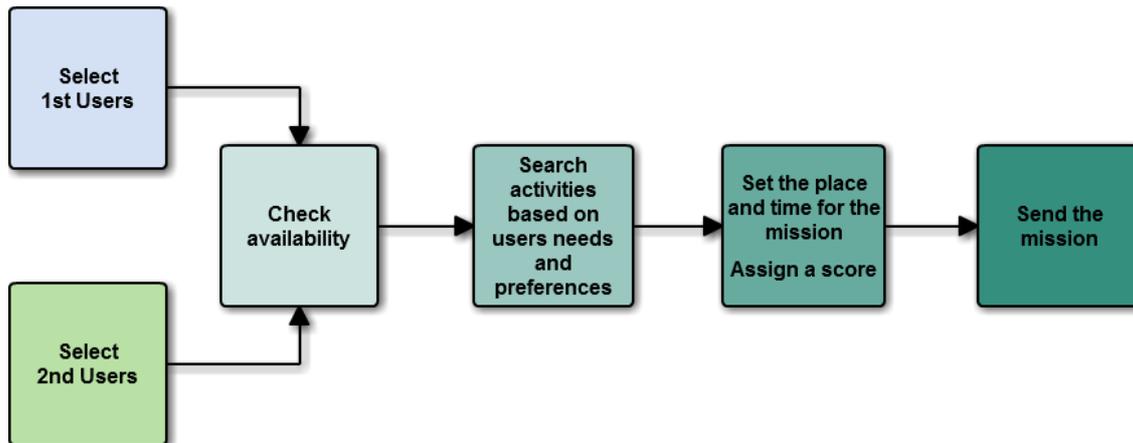


Figure 6 - Mission creation workflow

### Re-plan or cancel a rejected mission.

If a volunteer rejects a mission, a 3<sup>rd</sup> user of that care center can re-plan or cancel it depending on the rejection code reported by the volunteer.

- Re-plan: the 3<sup>rd</sup> user can change the date or time for the mission based on the 2<sup>nd</sup> user's feedback. The volunteer shall receive a new notification.
- Cancel: When the mission is cancelled the mission ends.

### Cancel from planned, started or finished

The 3<sup>rd</sup> user can manually cancel a mission that is in status Planned, Started or Finished, e.g. the 2<sup>nd</sup> user has called the care center to ask for the cancellation due to personal reasons. To cancel the mission the 3<sup>rd</sup> user needs to enter the reason for this action and request the system to cancel it.

### 2<sup>nd</sup> users workflow

#### Accept/Reject

When a 3<sup>rd</sup> user creates a mission the volunteer will receive an automatic message notifying this assignment. At this point the 2<sup>nd</sup> user can accept or reject the mission.

- Accepting a mission – The 2nd user confirms that he will perform the mission. The status of the mission changes to status Planned and stays in that status until the volunteer starts it on the scheduled date.
- Rejecting a mission – If the 2<sup>nd</sup> user rejects a mission (s)he should indicate the reason for rejecting it. The system will offer a drop-down list with default answers. Upon reception of this response, the 3<sup>rd</sup> user will be able to re-plan or cancel the mission.

### **Start/Request Cancellation**

The volunteers can start or request cancellations of accepted missions.

- Starting a mission – When the 2<sup>nd</sup> user is ready to start a scheduled mission, (s)he has to select the corresponding option (press the “Start” button) for that specific mission on the mobile application. This has to be done at the moment the volunteer starts the way to MyMate destination. The mission will automatically change to status Started.
- Requesting a cancellation of a mission – If the 2<sup>nd</sup> user cannot perform the mission, for some unexpected reason, (s)he has to call the care center and report that. The 3<sup>rd</sup> user will then be able to cancel the mission, which will change to status Cancelled.

### **Finish/Request Cancellation**

When a started mission is finished or needs to be cancelled, the 2<sup>nd</sup> user needs to take following actions:

- Finishing a mission – When the 2<sup>nd</sup> user has finished a mission, e.g. (s)he gets home, (s)he needs to select the corresponding option (press the “Finish” button) on the mobile application, answer the pop-up questionnaire, e.g. open questions form, that was customized by the care center and submit the filled-in form. The mission will change to status Finished and the 3<sup>rd</sup> user can make an evaluation of the mission.
- Requesting a cancellation of a mission - – If the 2<sup>nd</sup> user cannot complete the mission, for some reason, (s)he has to call the care center and report that. The 3<sup>rd</sup> user will then be able to cancel the mission, which will change to status Cancelled.

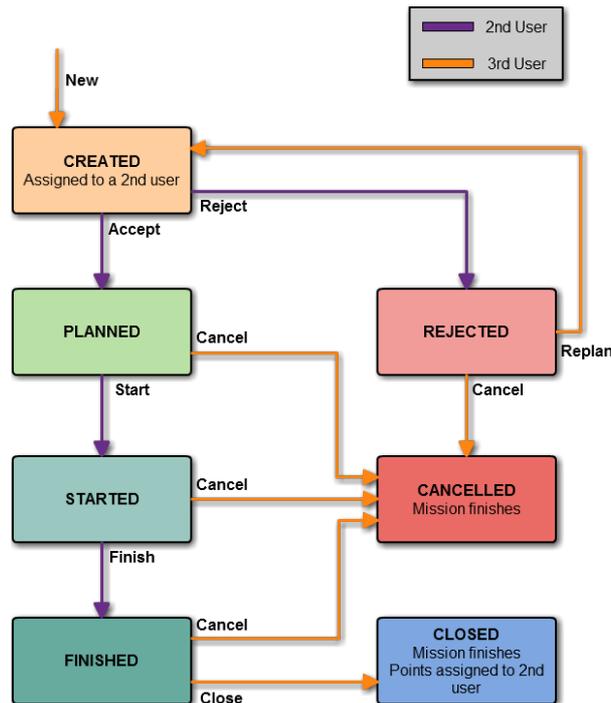


Figure 7 - Game mission progress workflow

## 7.2. Points

The point system will be the core of the scoring and reward modules.

- Each mission will have a number of points to reward the accomplishment.
- The accumulated points will define the level of the user.
- Each center will have its own ranking, using the total accumulated points obtained by each user, and there will be a global ranking of all users in the system.
- There will be a rewards system to exchange the rewarded points for gifts, which are defined by each care center.

## 7.3. Levels

There will be several levels or medals that will be awarded to the 2<sup>nd</sup> users depending on their accumulated points. Depending on the level, the same rewards will have a higher or lower cost, thus rewarding the users more committed to the program.

## 7.4. Rewards

The rewards system will be a section with a list of gifts offered by the care center and their cost in points. Only the rewards that can be afforded by the 2<sup>nd</sup> user will be shown, and the price will be different for each participant

depending on the current user level. Also, information on rewards can be shared on the registered social networks

- Exchanging the rewards – If a user selects to exchange some rewarded points for gifts, the rewards system will take the corresponding points from the user account (keeping the accumulated points defining the user level untouched) and generate a Bidi (QR) code for the user.
- The user will be able to use that QR code at the corresponding shop to pay for the granted reward.

On the other hand, the 3<sup>rd</sup> user can configure the rewards by means of CRU operations:

- The 3<sup>rd</sup> user will introduce following data when creating a reward:
  - What is the reward
  - Shops where the reward can be exchanged
  - Picture
  - Number of times a single user can get that reward
  - Price in points for each user level
- The allowed Update operation is to enable/disable the reward. If a reward is disabled it cannot be exchanged by the users

### 7.5. Rankings

Every care center will have an internal ranking. The 2<sup>nd</sup> users will be allowed to do the following operations related to the ranking:

- See the position in the ranking
- Send messages to other users
- See some authorized information of other users, e.g. points and alias.
- Share in Facebook his/her current position in the ranking

### 7.6. Assessment and reporting

The assessment and reporting system intends to evaluate the performance of the volunteer and the matching level with the 1<sup>st</sup> user and also receive feedback from the volunteer with regards to the mission, e.g. how they felt performing it.

Therefore, this system will have two parts:

- Reporting – when the mission is finished, the application will make a set of questions to the 2<sup>nd</sup> user. The user will answer selecting one of the pre-defined answers. This report will provide a general view of the mission. Optionally, the user can provide additional comments in an open field provided for this purpose.

- Assessment – Once the 2<sup>nd</sup> user has sent the form, the 3<sup>rd</sup> user will receive and analyse it. The next step will be to call the 1<sup>st</sup> user and ask him/her some questions about the mission. Then, the 3<sup>rd</sup> user will evaluate the mission and the performance of the 2<sup>nd</sup> user and will write a report. Finally, the 3<sup>rd</sup> user will close the mission and the system will reward the volunteer.

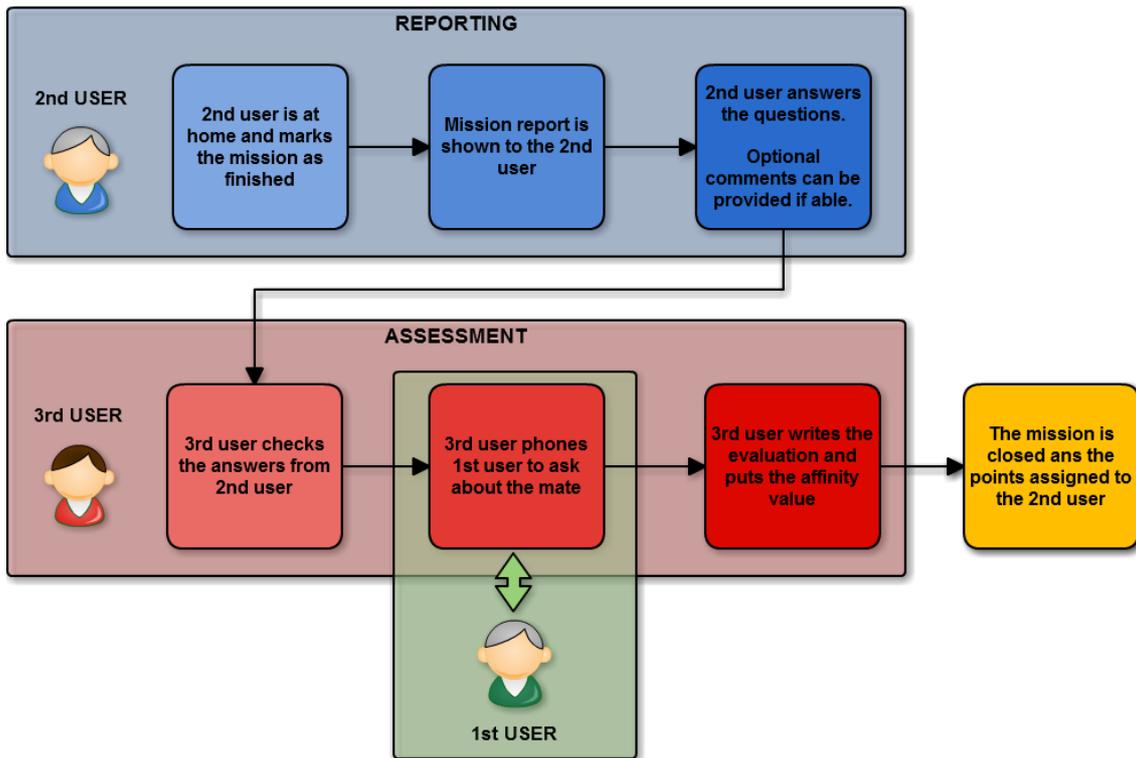


Figure 8 - Assessment and reporting workflow

## 8. STORYBOARD

This chapter presents the designs and contents for the different screens of the tablet application.

The design has been made taking into account the kind of users that will use the application, older people, which do not usually have too much experience with the new technologies. To accommodate the usability of the application to all levels of user skills, the user interface has been designed to be simple, intuitive and easy to use, e.g. the menus and controls will be visible and highlighted within the different contents presented. This is the first approximation of the designs, but could vary based on usability.

Each screen of the application will belong to a specific category based on the function fulfilled within the platform.

### 8.1. Login

The login screens allow the users to start the application (Figure 9). The users will be able to log in using any of the following accounts:

- MyMate account
- Facebook account
- Google+ account

Once the users are logged in, they will be able to see their accounts and configure the different parameters from the menu on the home screen (Figure 10). The menu gives access to the following parameters:

- Current mission: If there is any mission active, it will appear here.
- Mission overview: The user can view the available missions or the finished ones.
- Chat: The user can chat with other 2<sup>nd</sup> or 3<sup>rd</sup> users.
- Profile: Own profile overview and ranking.
- Rewards: Access to the rewards store.



Figure 9 - Login

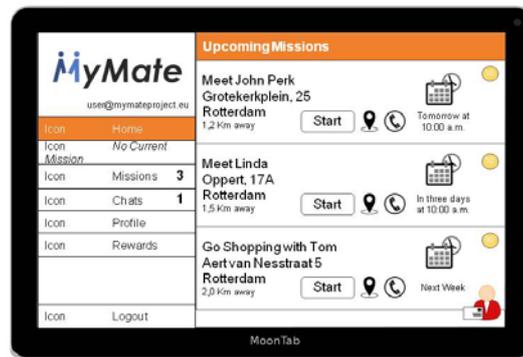


Figure 10 – User home

## 8.2. Mission income

When a 3<sup>rd</sup> user creates and assigns a mission to a 2<sup>nd</sup> user, the 2<sup>nd</sup> user receives a push notification (Figure 11) and the mission is moved to the Inbox (Figure 12) section.

When the 2<sup>nd</sup> user clicks on the mission, the mission is unfolded and all the details of the mission are shown. Also, two options will appear, allowing the user to accept or decline that mission (Figure 13). If the user accepts the mission, it goes to the planned missions. Instead, if the user clicks the button Decline, a dialog is shown (Figure 14) because the user must provide the reason for the rejection.

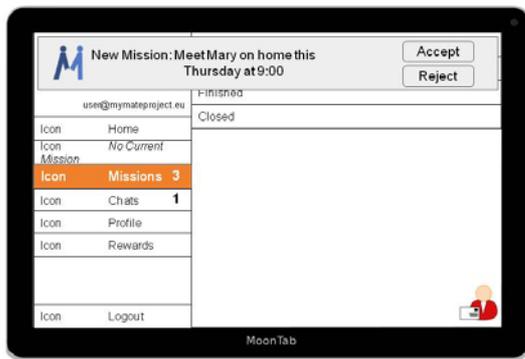


Figure 11 - New mission notification

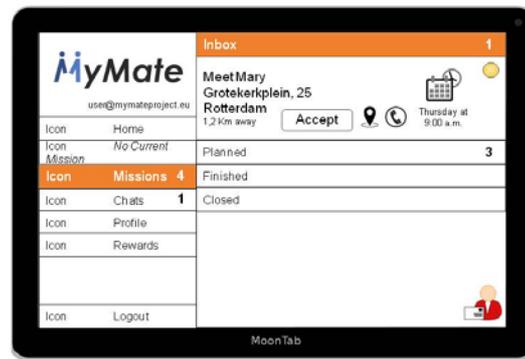


Figure 12 - New mission list

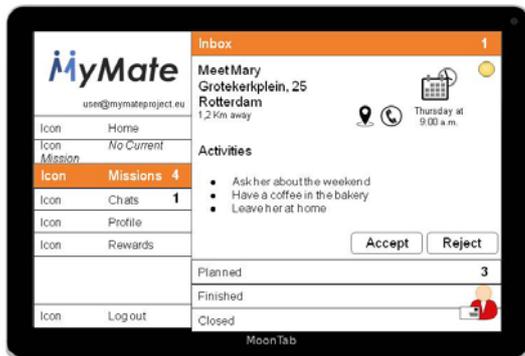


Figure 13 - New mission detail



Figure 14 - Mission decline dialog

### 8.3. Mission overview

On the mission menu (Figure 15) the user will have four options:

- Inbox
- Planned
- Finished
- Closed

Once a mission is accepted, the user can access it on the planned missions menu (Figure 16). This menu shows all the missions accepted by the user and pending to be started on the scheduled date. On the day arranged to perform the mission, the user should click over the Start button to begin the mission.

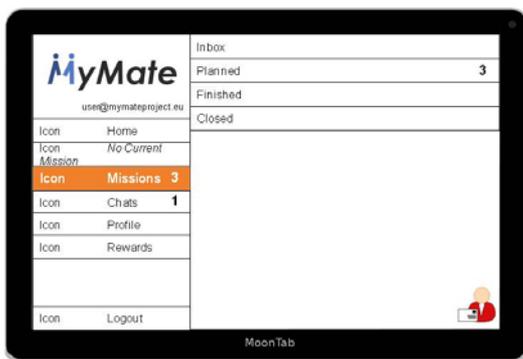


Figure 15 - Mission list menu

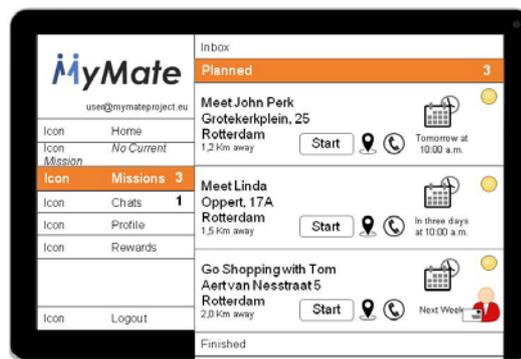


Figure 16 - Pending mission list

### 8.4. Mission accomplishment

Once one mission has begun, the 2<sup>nd</sup> user must go to the 1<sup>st</sup> user’s address and check the corresponding button when he meets his mate (Figure 17).

After that, (s)he should accomplish the different tasks related to the mission. When all planned tasks have been completed the mission can conclude. The 2<sup>nd</sup> user will need to press the “Finish mission” button (Figure 18) to conclude the mission.

Later on, when the mission is over, the 3<sup>rd</sup> user tracking it will contact the 1<sup>st</sup> user and ask for feedback.

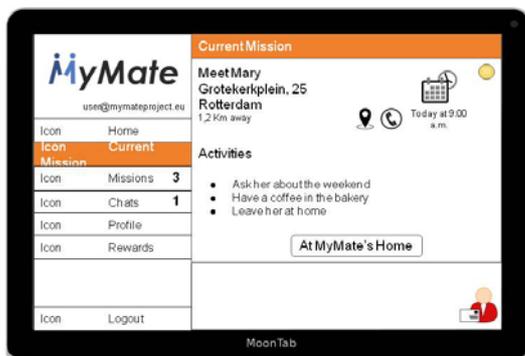


Figure 17 - Mission starting

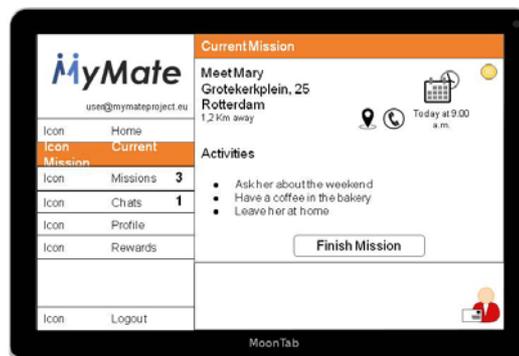


Figure 18 - Mission ending

### 8.5. Mission report

When a mission is finished, the 2<sup>nd</sup> users must fill in a form to report the results of the mission (Figure 19 to Figure 22) and give their opinion about the mission in the form of a predefined questionnaire. Also, the 2<sup>nd</sup> users will be asked to send a form providing their own free comments of the mission.

Once the user sends the form, the mission goes to the finished missions section and gets pending the validation of the 3<sup>rd</sup> user until the 3<sup>rd</sup> user verifies the correct accomplishment of the mission.

On the finished missions screen (Figure 23) the user can see all the missions that he has finished, but are pending the approval of the 3<sup>rd</sup> user, who should check if the mission has been completed correctly.

The closed missions screen (Figure 24) shows all the missions than have been completed and checked and verified by the responsible 3<sup>rd</sup> user at the care center. In that moment, the points are granted to the user and can be used in the rewards manager.

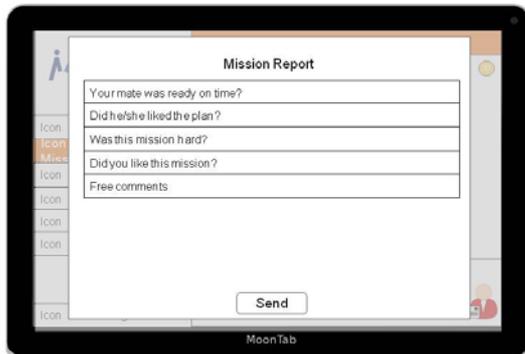


Figure 19 - Finish mission report

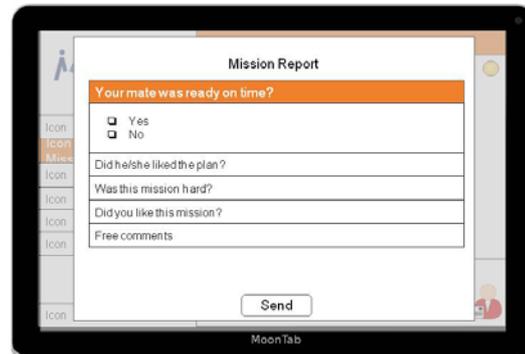


Figure 20 - Finish mission report

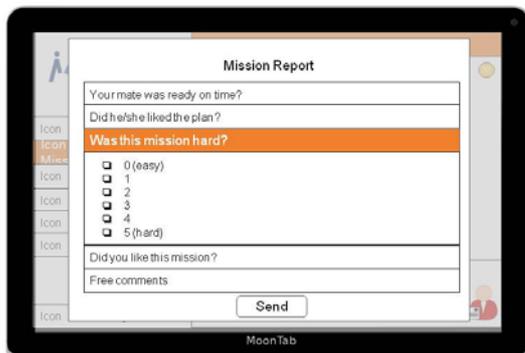


Figure 21 - Finish mission report

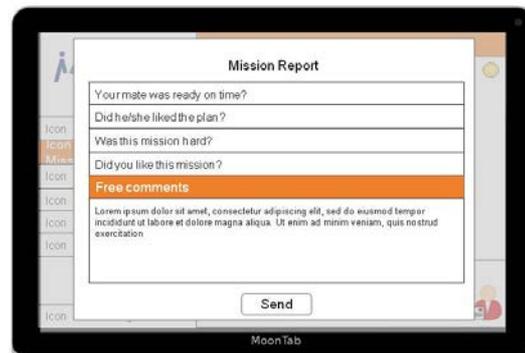


Figure 22 - Finish mission report



Figure 23 - Mission finished

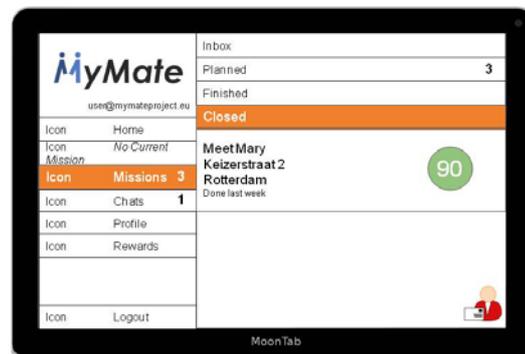


Figure 24 - Mission closed

In every screens the user can click on a mission and see all the details about it.

## 8.6. Profile management

In this section the 2<sup>nd</sup> users can view their personal details and review their calendar of availability (Figure 27) but, for making any change affecting the calendar information, the users must call the care center. The days the user is available will be marked in bold.

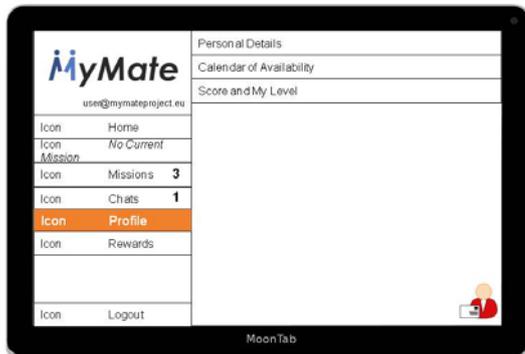


Figure 25 - Profile menu

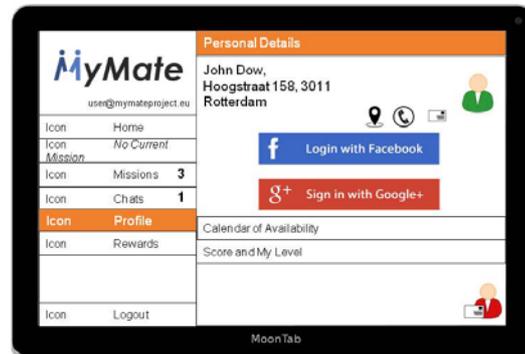


Figure 26 - User profile information

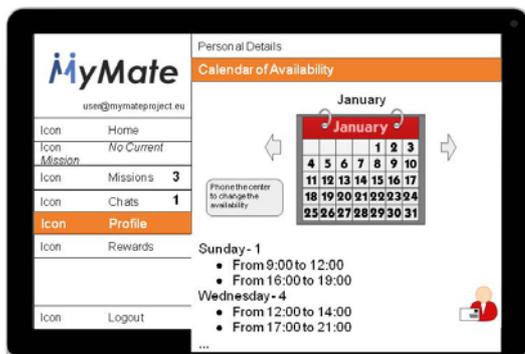


Figure 27 - User availability

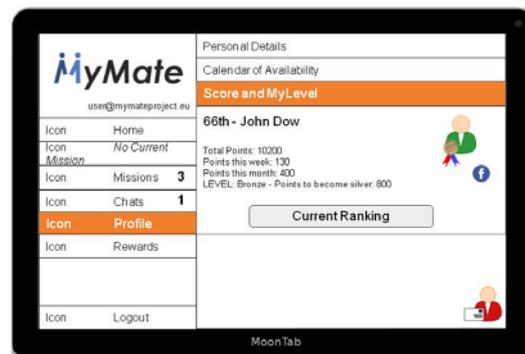


Figure 28 - User score

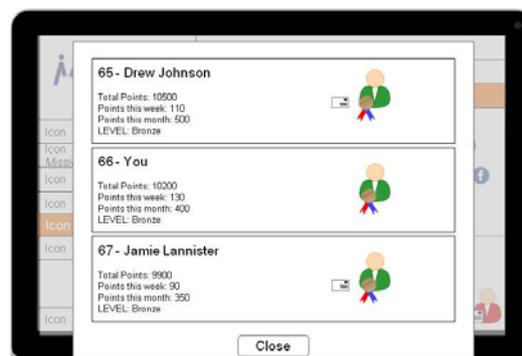


Figure 29 - User ranking

In the user profile screen (Figure 26) the user can update his/her personal data and use his/her social accounts for login into the application or sharing information.

Also, in the user score screen (Figure 28) the user can view his/her current score, user level and earned MyMate points. The user can share the score, and view the global and local (within the care center) ranking by selecting the Current Ranking

option (Figure 29). The user can watch his/her position in the ranking and scroll up and down the list. The user will be able to send messages to other 2<sup>nd</sup> users by clicking on the mail icon.

### 8.7. Chat

This list shows all the chats of the user. The chats are ordered by date and the newer ones go first. A bold number shows the number of unread messages on each chat.

The 2<sup>nd</sup> user can send messages to 2<sup>nd</sup> or 3<sup>rd</sup> users. When a chat is selected it opens and the messages are marked as read.

The chat works as a common chat application to allow using it with no special training.

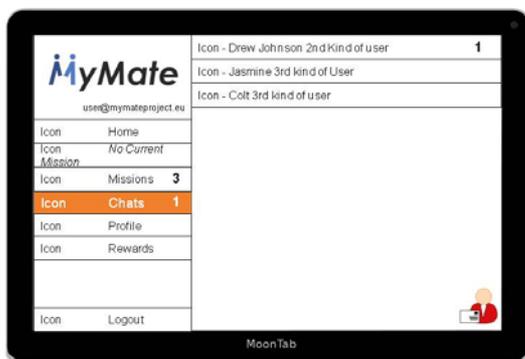


Figure 30 - Chat menu

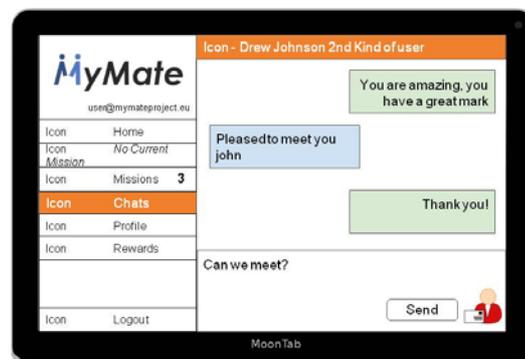


Figure 31 - Chat

### 8.8. Rewards manager

This screen shows the available rewards. The rewards the user can afford to exchange using his/her current points are displayed in green, the rest of rewards will be grey.

Depending on the kind of user, e.g. bronze, silver, gold, platinum and diamond, the reward will have a different price in points. The price for the current user will be underlined with a green line.



Figure 32 - Rewards manager