Senior Design Project

S.U.N (Social University Network)

Analysis Report

Group Members

Ahmet Ertan

Basri Karademir

Damla Özgen

Dilan Yatan

Yeliz Kurt

Supervisor

Can Alkan

Jury Members

Hakan Ferhatosmanoğlu

Çiğdem Gündüz

Innovation Expert

Mehmet Çakır

TABLE OF CONTENTS

Introduction

2. Proposed system

2.1 Overview

2.2 Functional Requirements

2.3 Nonfunctional Requirements

2.4 Pseudo requirements

2.5 System models

2.5.1 Scenarios

2.5.2 Use case model

2.5.3 Object and class model

2.5.4 Dynamic models

2.5.5 User interface - navigational paths and screen mock-ups

3 Glossary

4 References

II. Table of Figures

Figure 1: Use Case Model of the System

Figure 2: Class Diagram of the system

Figure 3: Log-in Sequence(Client Side)

Figure 4: Log-in Sequence(Server Side)

Figure 5:Sequence diagram of create event (client)

Figure 6:Sequence diagram of create event (server)

Figure 7: Login Page

Figure 8: Profile Page

Figure 9: profile subpages

Figure 10:Main Page

Figure 11: pop up1

Figure 12: pop up2

Figure 13: Events Page

Figure 14: Search Friends

Figure 15:other user's profile

Figure 16:search interests

Figure 17:interests details

Figure 18: Create Events

Figure 19:Event Details

1. Introduction

University is a place where students experience their final learning stage before the professional business life. Moreover, students find most of their contacts in those years. Unfortunately in crowded universities, socializing is one of the main problems. People mostly tend to keep in touch with the people they have already known since the first years of their education. After first or the second year, socialising with new people gets harder. Especially in engineering or science departments where students are less willing to socialise.

College environment is the most efficient chance to solve the socialising problem. In order to solve the problem for college students, a social network had been thought to encourage them to meet new people in their university life easily. The main idea of this project is make people work together and participate events together even though they do not know each other by making the situation free from burden of trying to introduce themselves personally at the beginning. With the help of technology, our system carries this hardest, problematic stage to the mobile environment, decorates it with competition and presents it as a way of having a joyful college life.

For having fun, students can create or join some hobby events like snow ball figting of movie nights. To be a more active colleger, students can follow the student clubs which are member of our system and attend their events. Also, being a student requires hard working in Bilkent, therefore our system let's students create study groups as a study event for everyone who interested in topic to attend.

Attending events, meeting new people, working with them, spend time in the campus together, especially creating own events gives points to the users for them to earn badges and raise their ranking in the points competition.

The project is an Android application which is designed for mobile platform, called “S.U.N” (Social Student Network). The students will be able to download the application to their smart phones and start to challenge. The initial target group will be the İhsan Doğramacı Bilkent University students. They need to register to the system before using the application. While registering, all they need are their university-mail, full name and a password for their security. After signing in, they may choose their courses, hobbies or clubs from the system’s interest list. Main Pages are user specific. Those interests they chose will help us to know which events to inform to the user on their Main Page. Also they will be able to add friends to being informed about which events they are joining or compete themselves with those friends.

To get points, user should choose the join an event by touching the join button under event page and when the event is occuring, user should bump his phone with the event creator or someone already Bumped his phones with the event creator. Users can directly go to the event and Bump the phones but sure it will give them less points because one of the basic property that a college student must have is being organised.

This report will give more detailed overview of our project, it will describe the functional and non-functional requirements and it will depict our software with all necessary system models and its user interface.

2. Proposed System

2.1 Overviev

Our project, SUN application, will be an Android mobile application. The purpose of the project is to encourage university students to be socially active in college environment. To use the application, users have to create profile and login to the system. It will be a check-in based system that uses NFC and Gsm technology included by the Android phones and users get points and badges after their participation to the events controlled by this check-in system. Gsm controls if the user is near the area and NFC controls a more accurate knowledge if the user is really attended to the event. Also this property will help the application to become popular when a student arrives to some

event area in school and asks “Do you have s.u.n loaded in your phone” in order to bump phones. Also it adds fun to the user experience.

Main functionality of the application is determining events and providing check-in system for those events. User scenarios and class model show the system structure in a more detailed way.

Technical Details of S.U.N

Platform: Android smartphones

Language: JAVA

Database: MySql

2.2 Non-Functional Requirements

→ Users will be able to register and log in to the system.

→Users will also be able to register all existing courses hobbies or cl in the university.

→ Users will be able to create events for a course, or hobby.

→ Users will be able to edit or cancel the created events..

→ The system will notify the user about upcoming events or the news about university.

→ Users will be able to complete challenges created in the system.

→ The application provides a search and a sorting function for all entries.

→ Users will be able to check in. System will control the accuracy.

→ Basic information and already created events about a registered interest of course, university club or hobby can be seen by users who registered that interests.

→ Users can apply to have a University Student Club like Mac Business or IEEE in Bilkent to have a special account in order to create club activities.

2.3 Non-Functional Requirements

→Eric, logs into his account with his e-mail.

→He wonders what events he attended, and he touchs on “Past Events” button.

→Then he sees, his events, that some of them is created by him and some of them is created by other users.

→Suddenly, he remembered his Calculus midterm but he couldn’t remember the event of Calculus Midterm study time. So he touchs on “Planned Events” button, to see the time of the event he created.

2.4 Pseudo Requirements

→ The program has to be implemented in Java.

→ The client has to be a mobile application.

→ The database has to use Structured Query Language (SQL)

2.5 System Models

2.5.1 Scenarios

Scenario name: **createAccount**

Participating actor instances: Eric: User

Flow of events:

1. Eric is a Bilkent 3rd grade Bilkent student who has enough of friends but wants to enjoy the college life more. So he opens S.U.N application he already installed on his Android smartphone.
2. He wants to sign up, so he enters his information and be logged in automatically.
3. After registration, because of this is Eric's first timein the application , Eric is redirected to his Profile page and and saw his interests list is empty.
4. To ass some interests to his profile, he chosed between enjoyable or profitable interests which can be performed in his college.

Scenario name: **login**

Participating actor instances: Eric: User

Flow of events:

1. Eric starts the application to see if there are interesting news or events upcoming.
2. When the login page appears, he logs in by entering his school e-mail, and the password he defined

3. He is redirected to the home page of the application and he sees the News Feed.
4. He sees a warning at the News Feed which is about the deadline of the withdraws.

Scenario name: joinEvent

Participating actor instances: Eric: User

Flow of events:

1. Eric was alone in a coffeeshop and feels bored. He logs into application by entering his school e-mail.
2. In the home page of the system there are some events, that created by other users.
3. He touchs on CS101 study event .There is location and time of the workplace in the information part of the event.
4. He joined on CS101 study event. Now the event is in his event list.
5. Then he remembers that there is Calculus Midterm on saturday morning and he checks the home page if there is an event for studying Calculus.
6. He saw events related, but the time and date was not convenient for him. He decided to create a new one.

Scenario name: **createEvent**

Participating actor instances: Eric: User

Flow of events:

5. Eric has a calculus exam and his time schedule is not fitting with any of the existing events.
6. He thinks that there might be another students like him and wants create a new event abot calculus exam.
7. He loggs into the application, from the events menu, select create new ewent.
8. He is redirected to the interests menu, he choose calculus course 121 between courses and enters the event creation page.
9. he decided on date,time and place.
10. Eric saved his event. Five minutes later, eric had a notification about his first participiant.

Scenario name: **checkEvents**

Participating actor instances: Eric: User

Flow of events:

1. Eric was talking to his friends how helpful the S.U.N Application is and he wants to show what events he has attended. He opens his application.
2. He enters his profile and he touchs on “Past Events” button.
3. Then he sees, his events, that some of them is created by him and some of them is created by other users.

4. He brags about ho social he is.
5. Suddenly, he remembered his Calculus midterm but he couldn't remember the event of "Calculus Midterm study"s time. So he touchs on "My Events" button, to see the time of the event he created and saw it is 15 minutes later.
6. he starts to run to the library while thinking how bussy the college life is.

Scenario name: getStudentGroupAccount

Participating actor instances: Brandon: User

Flow of events:

1. Brandon is the concealor of a very famous student club in Bilkent. As a normai user of the S.U.N system, he knows how efficient to spread events from this application.
2. From the "contact us" link of the application, he sends the project team an e-mail about his club.
3. When the team checks if his informations are valid, he receives a mail about his student club account and now he is able to create club events.

2.5.2 Use Case Models

The main abilities of the user on this system is to check , create and attend events. User will also be able to learn about school news. Use Case diagram demonstrates the basic correlation between the actors and the functionality of the mobile application and shows the functional behavior of the system.

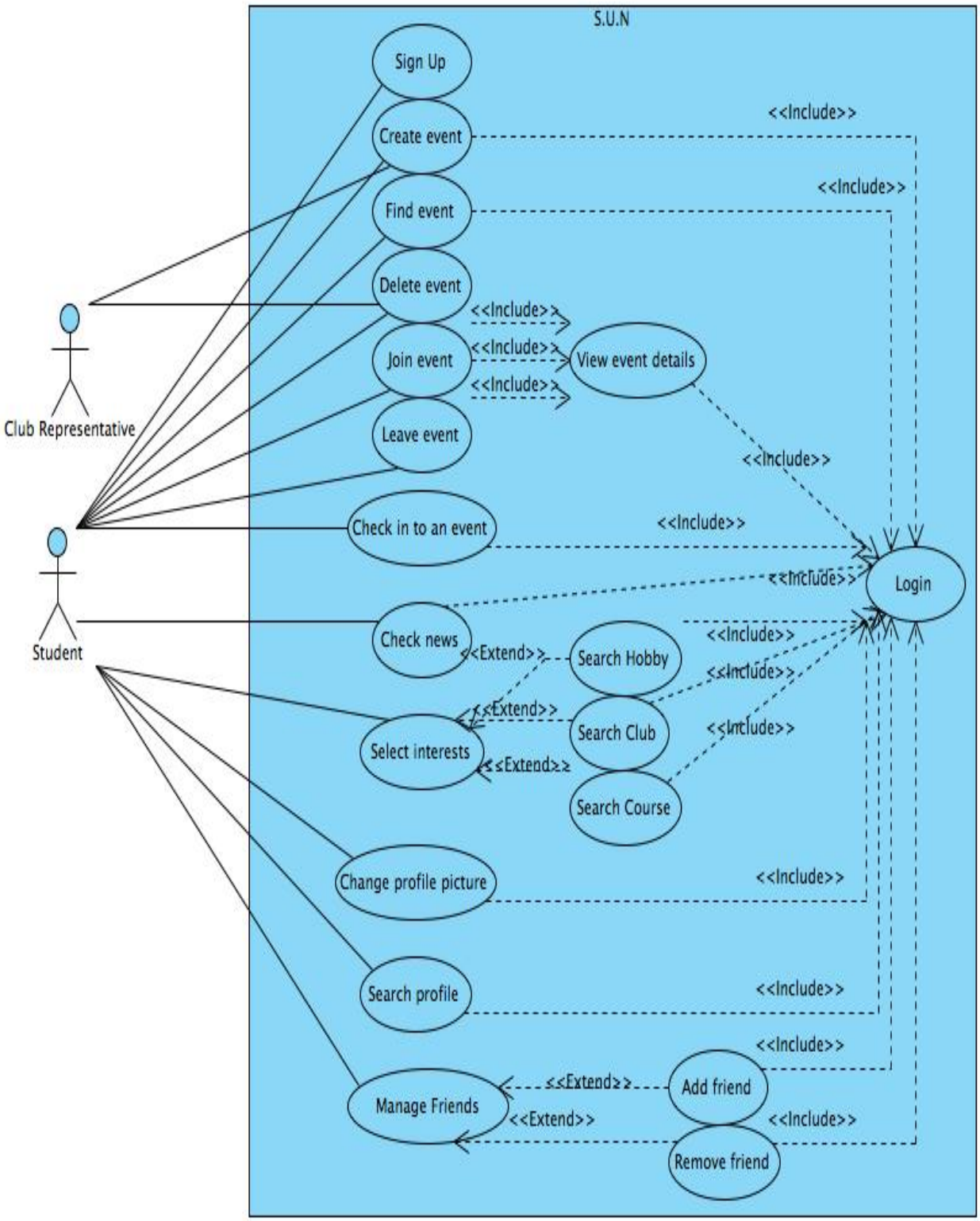

Figure 1: Use Case Model of the System

2.5.3 Object and class model

In this section the detailed class diagram of the project is shown. All the classes and functions are determined as detailed as possible to make sure that the implementation is easy and well-ordered. classes are designed to make the project object-oriented friendly.

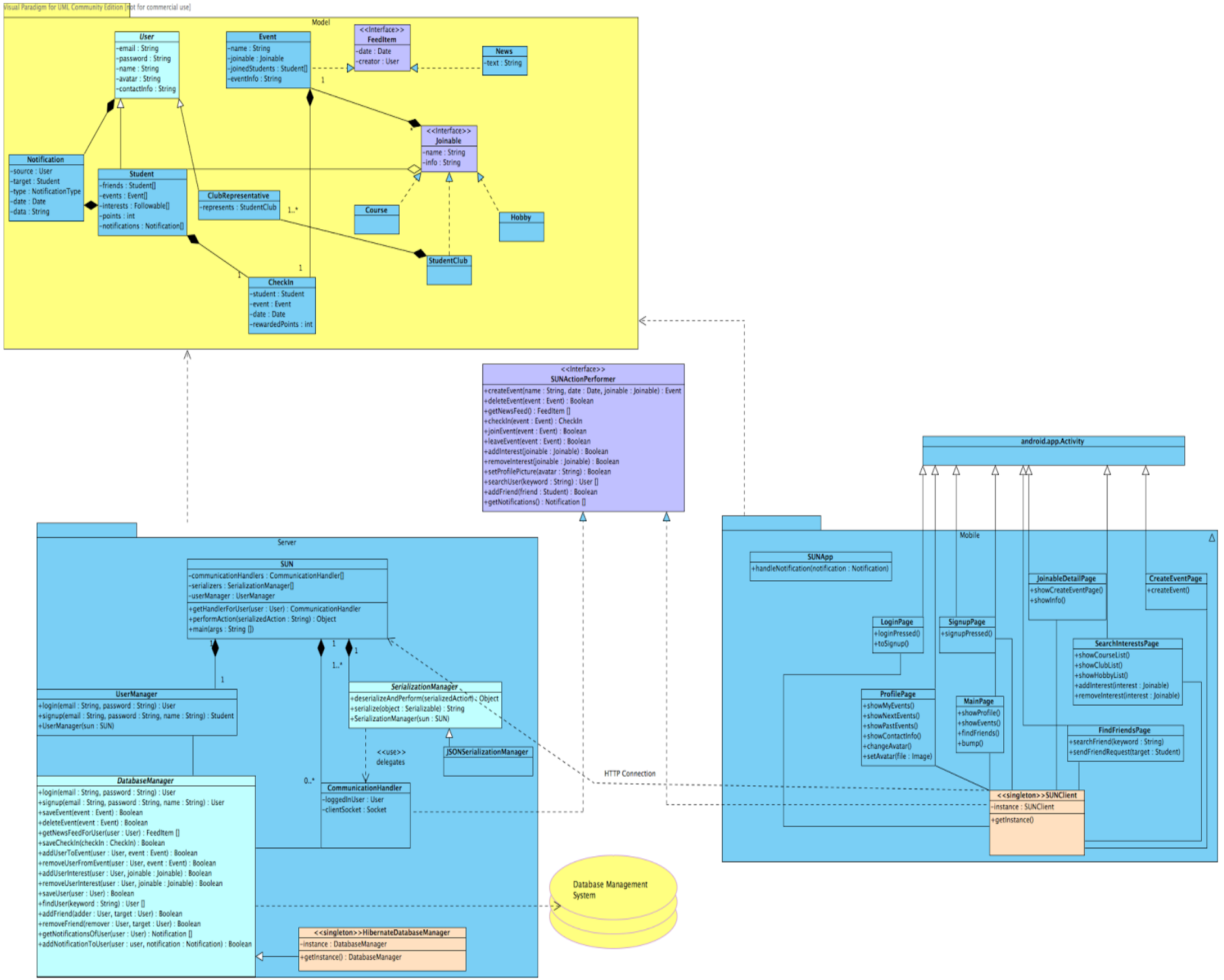

Figure 2: Class Diagram of the system

2.5.4 Dynamic models

In this section the more visualized form of how the system will think and act upon certain situations, the algorithms of the use cases shown before will be displayed. Sequence, State and Activity Diagrams are in this part. The details of "game plan" of the project like which functions are used when, can be observed via these diagrams.

2.5.4.1 Sequence Diagrams

2.5.4.1.1 Log-in Sequence(Client Side)

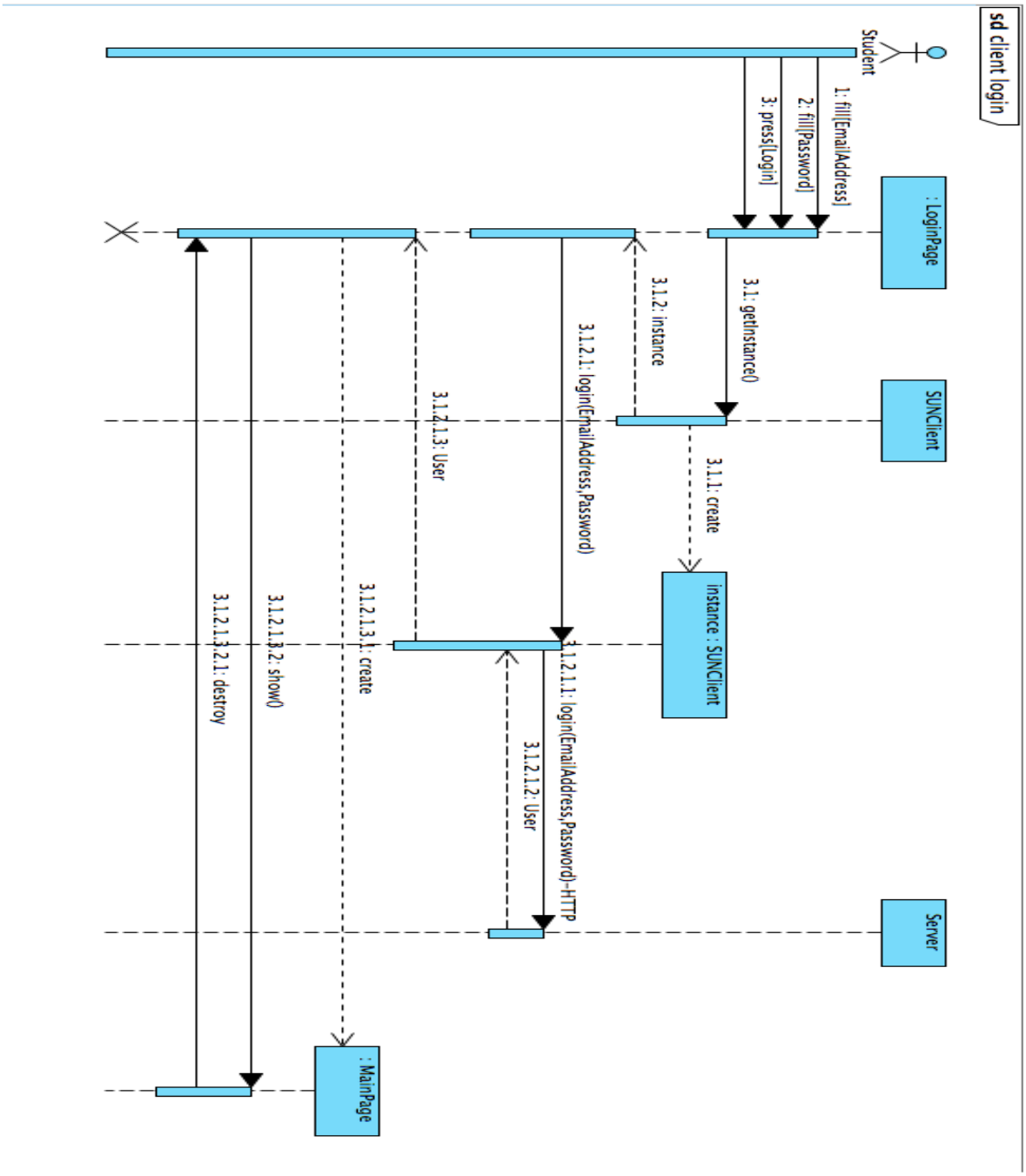

Figure 3: Log-in Sequence(Client Side)

This sequence starts when the user enters his credentials and press the login button. Then the screen retreives the common client object. After getting the object, the

screen calls it’s login method. Login method serialises the request into Json format, and send the request to the application server using HTTP connection.
When the server answers the request, this object desrialises the Json respond sent by the server and returns the response object to the caller. Since the response on the given scenario is true, login screen creates and shows the main screen.

2.5.4.1.2 Log-in Sequence(Server Side)

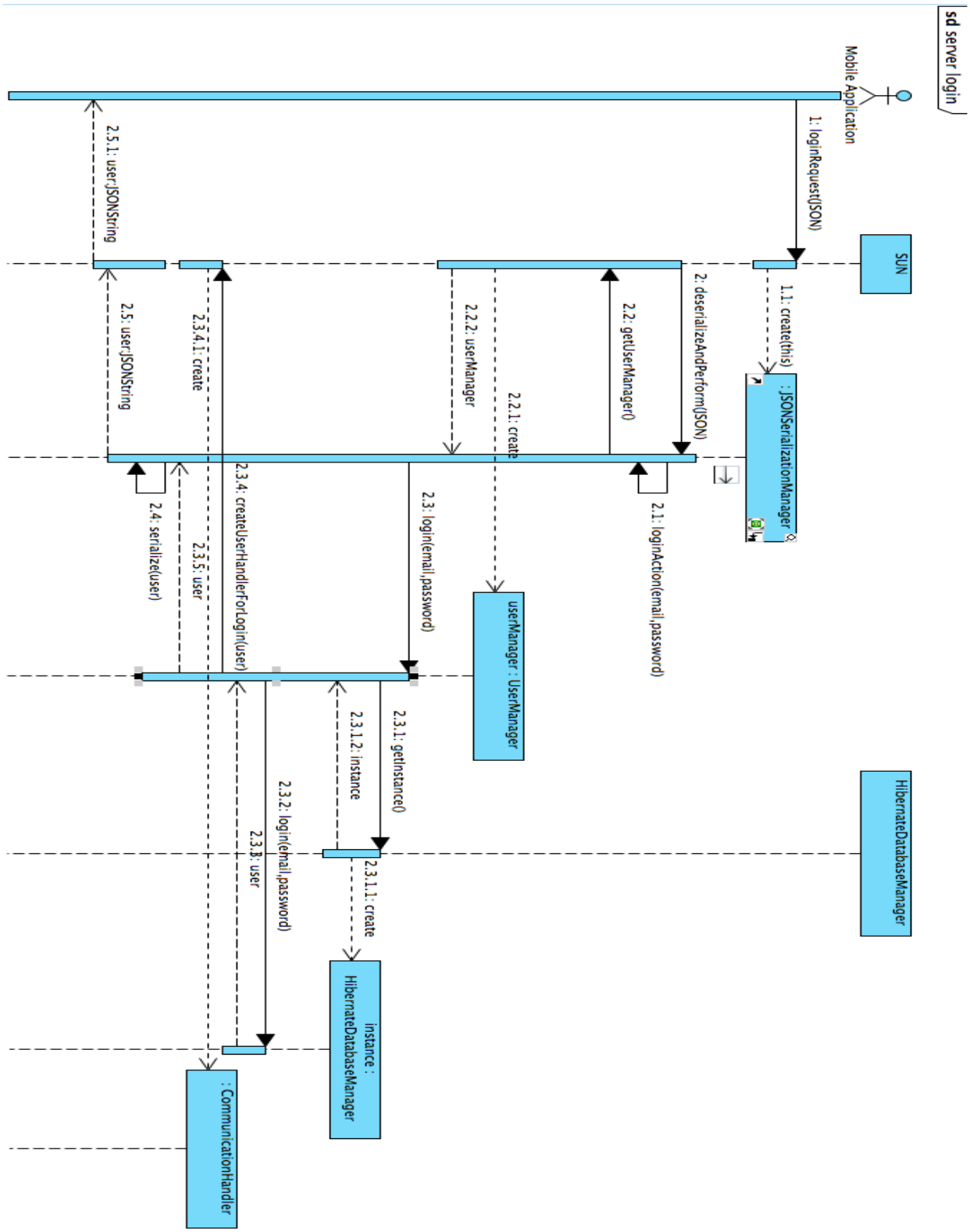

Figure 4: Log-in Sequence(Server Side)

This is the sequence of server's responding to an incoming login request. The request arrives at the main of the server component. Then the main method checks the format of the request and decides that it is in JSON format. After determining the format, it delegates the request to the instance of JSONSerializationManager. This manager parses the arrived request and understands that the request is a login request and gets the UserManager instance from the main class. Then checks the validity of the users credentials through UserManager instance. UserManager also delegates this request to DatabaseManager. After the validation, UserManager informs main class and the main class creates the ConnectionHandler object for the user. This object keeps the client socket as well as the user information and it will be used for the communication with that client until it is disconnected. Then JSONSerializationManager serializes a login successful message and returns it to main class. Then the main class responds to the incoming http request.

2.5.4.1.3 Create Event Sequence(Client Side)

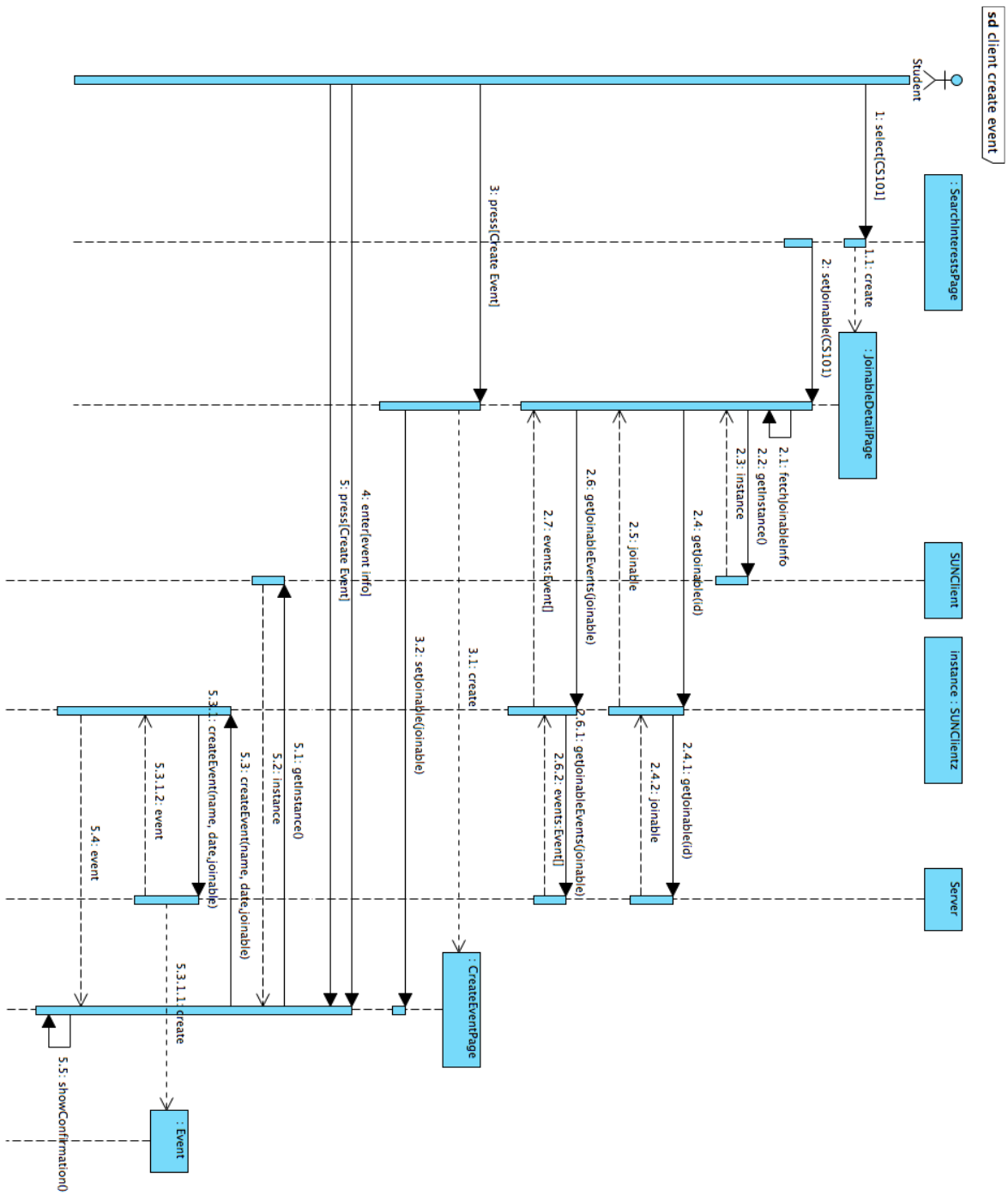

Figure 5:Sequence diagram of create event (client)

This diagram shows the sequence of events happening at the client application while the user is creating an event. First of all, the user selects a course from the course list. This action triggers the creation of the JoinableDetailPage. After getting created, the detail page requests the details of the course through SUNClient. In order to use the client, details page gets the shared instance of the SUNClient. Then it calls the getJoinable method of the client, this method sends an HTTP request to the server and parses the response into a usable model object as explained in the first sequence diagram. After getting the details of the course, the mobile application sends another request to get the active events of the course. This request also goes through the SUNClient object.

The user of the application then presses the “Create new event” button. As a result of this action, detail page creates a CreateEventPage object and shows it. Then the user enters the necessary information to the screen and presses the create button. CreateEventPage also uses the shared object of the SUNClient class and invokes the createEvent() method of it. Then the shared object serializes the request and sends to the server. Finally, the server sends a confirmation message and the create screen show the message to the user.

2.5.4.1.4 Create Event Sequence(Server Side)

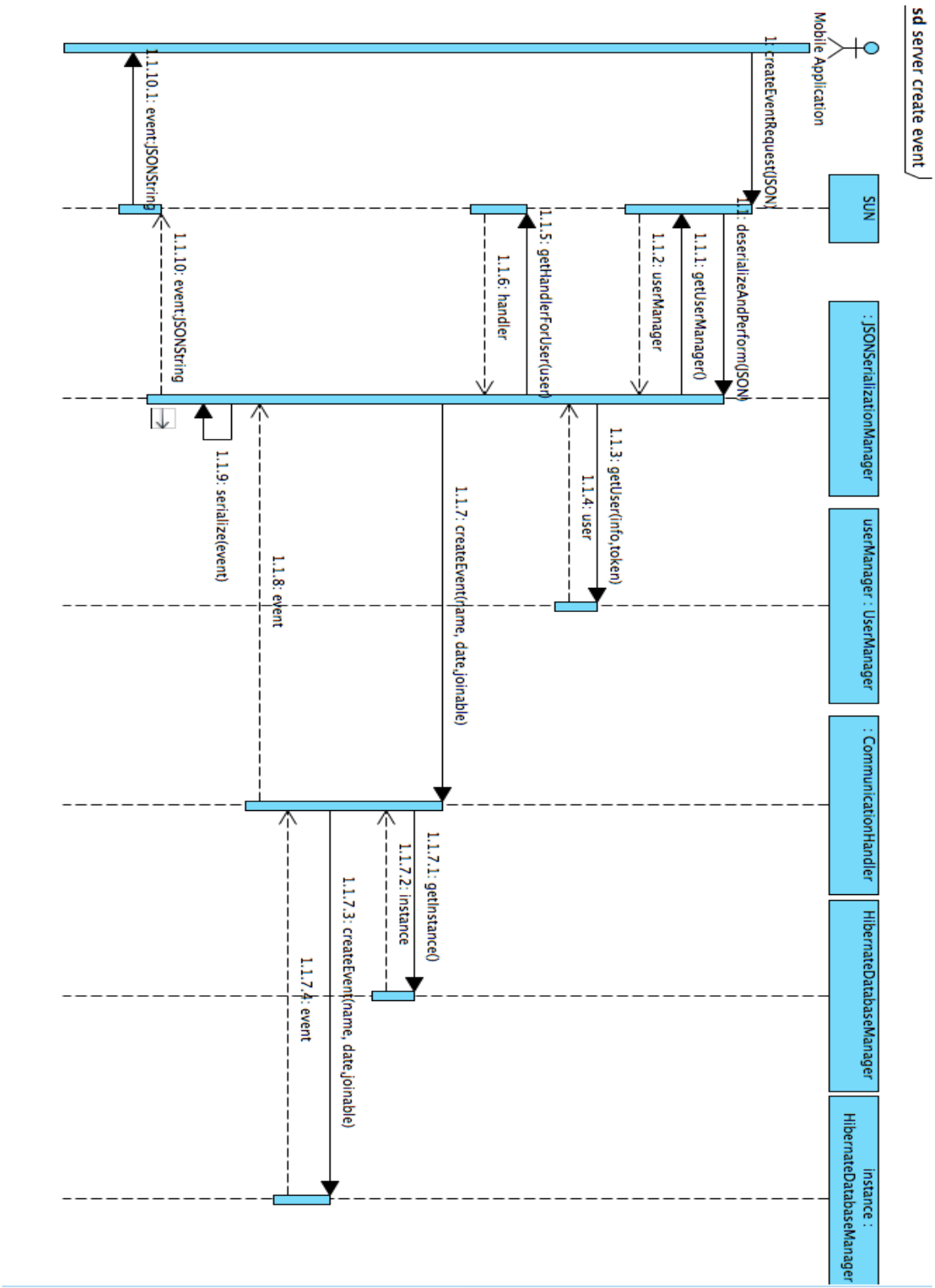

Figure 6:Sequence diagram of create event (server)

The sequence shown in the diagram starts after the create event request arrives at the server. Main method of the server directs the request to the JSONSerializationManager. JSONSerializationManager realizes that the request is a create event request and asks the main class for the user's ConnectionHandler. After getting the handler, the JSONSerializationManager delegates the request to the ConnectionHandler object. ConnectionHandler directs this request to the HibernateDatabaseManager class' shared instance and returns the result to the JSONSerializationManager. JSONSerializationManager then serializes the response to JSON and returns the generated string to the main class to be returned as the response of the HTTP request.

2.5.5 User Interface/Screen mockups

2.5.5.1 Log-in, sign-up pages:

Due to the fact that sign up is required in Social University Network system,when user starts the mobile application, the "Login" page welcomes them. There, user can Log in via his university-mail and his password. If he haven't created an account yet, by touching the sign-up link, he will be directed to the sign-up page where he can create one via entering his name, surname, university mail. Moreover he must define a password for his account. By touching the sign-up button, he will be signed up and automatically be logged in.

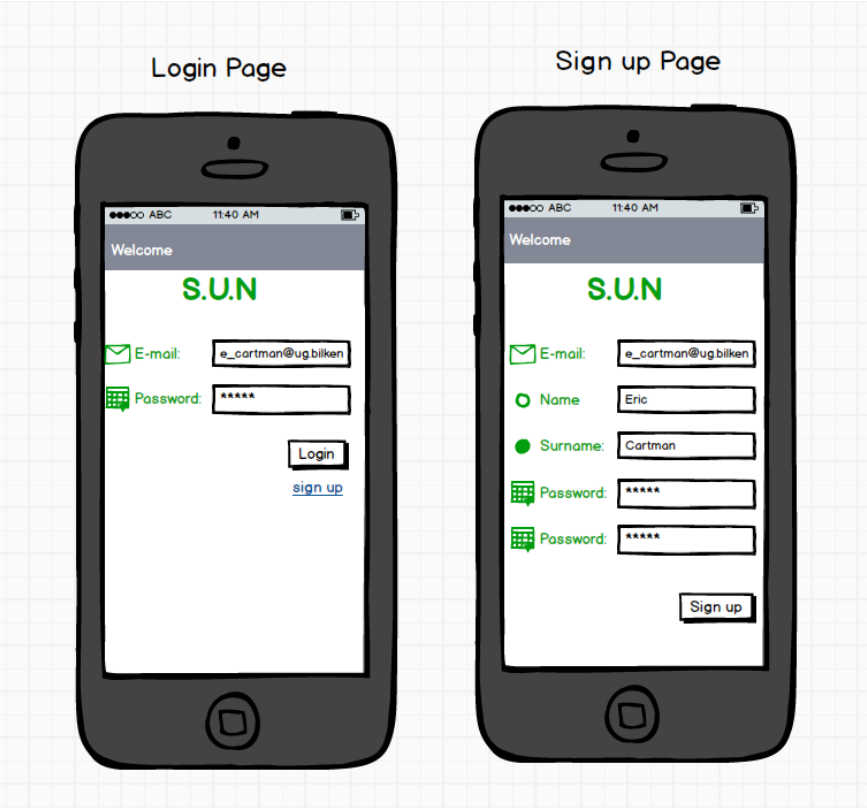

Figure 7: Login Page

2.5.5.2 Profile page

Normally, when the user logs in, he will be redir ected to the main page. Yet, if it's the first time he uses the app or if he is newly signed up, he will be redirected to the "Profile Page" basicly for him to set his profile picture and select his interests.

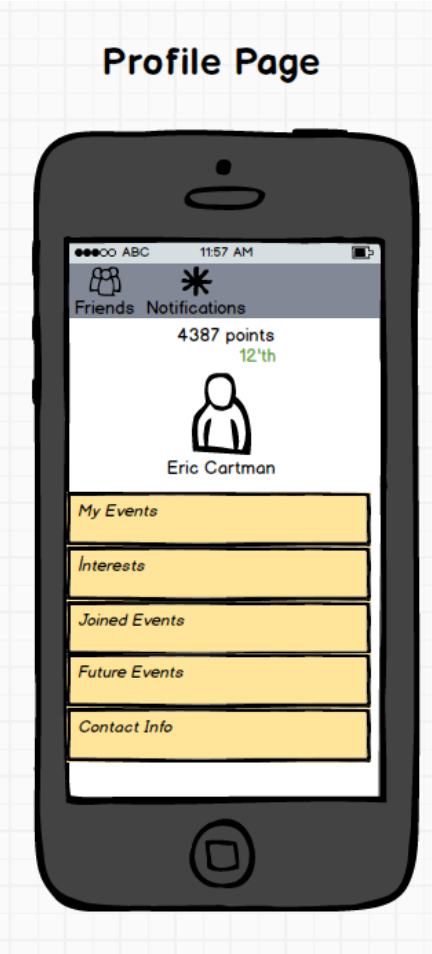

Figure 8: Profile Page

The Profile Page contains Buttons which are "My Events", "Interests", "Joined Events", "Future Events" and "Contact Info". Each of them will redirect the user to a corresponding pages shown below.

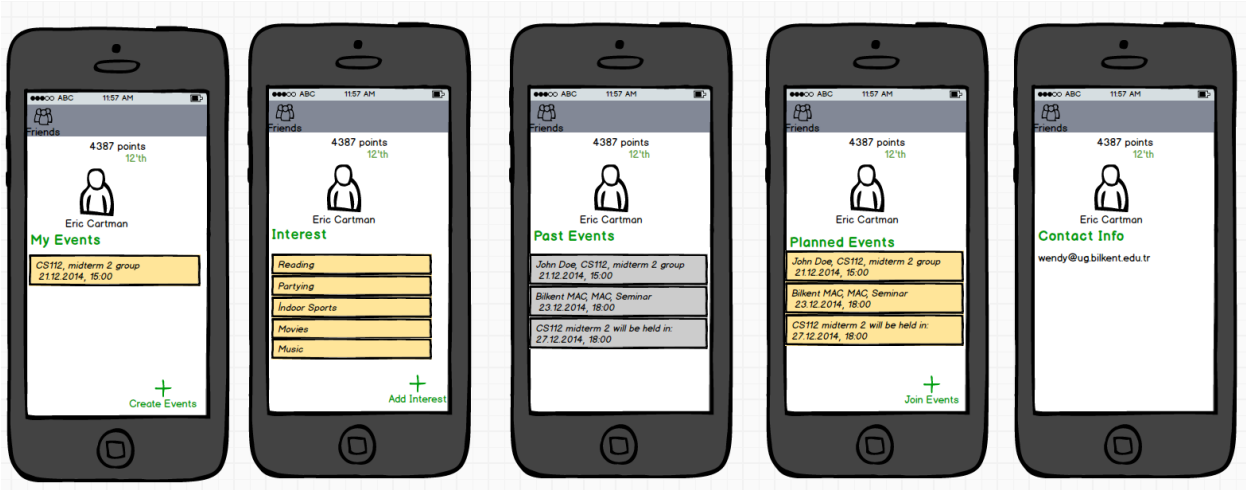

Figure 9: profile subpages

In “My Events” page, user will look up the events that he created. By touching the button “Create Events”, user will be directed to “search interests”page (3.5.5.6) in order to select which of the interests his event will be about.

“Interests” page will show the users selected interests. Those selections are used in the “News Feed” system to show the upcoming events that user might be interested. To add more interests, user can touch “add interests” button that leads him to “search interests” page (3.5.5.7) .

“Past Events” page one of the pages that can be reached from the Profile page, contains the events that the user joined.

“Planned Events” page contains the events that the user is planned to join (by touching the “join” button on the corresponding event’s page).

“Contact Info” contains the university-mail of the user.

From those five subpages,by touching the back button of the Android phone, user will be leaded back to the Profile page. From the Profile page, user will be redirected to the Main Page by touching the back button.

2.5.5.3 Main Page

After completing the log-in phase, user will be redirected to the main page which contains “News Feed”, “Profile” button, “events” button, “find friends” button and “Bump” button. In News Feed, there will be upcoming events about user’s “selected interest” which is explained in section 3.5.5.2. The upcoming interests text will be: “Name_of_creator , “name of interest, title, date, time”. Also, some news about the university. The news will be in a more pale color in order to ease the distinct them from upcoming events. If the user does not selected any interests, there only the news about the university will be shown.

+

Figure 10:Main Page

“Profile” button redirects the user to the profile page.

Events button will lead the user to the Events page (3.5.5.4), “find friends” button redirects the user to the Find Friends page (3.5.5.5) and the “Bump” button opens a pop up:

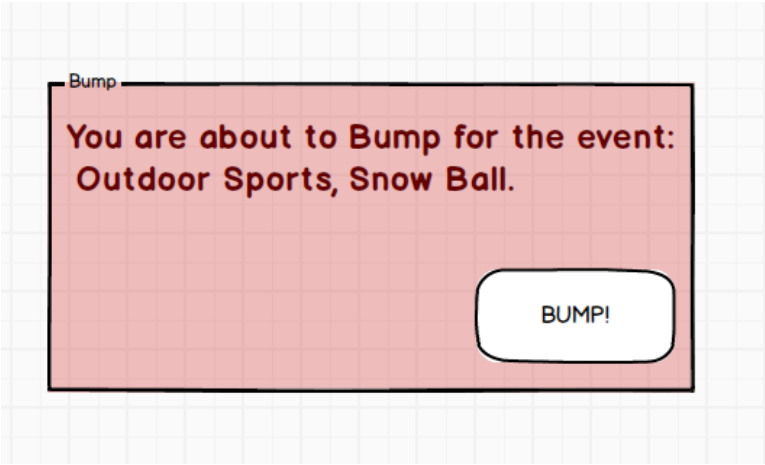

Figure 11: pop up1

The Application will lead user to “Bump” to get points of an event ,which is happening at that time , between the events user decided to join.

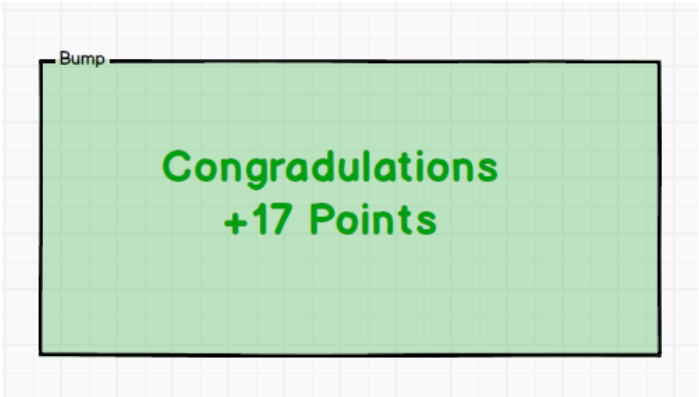

Figure 12: pop up2

If the user is successful to “Bump” phones with the event leader, the RF Id card or with someone already Bumped phones with the group leader or the Rf Id card, he will get points. The “Bump” actions can only be done near the event place.

2.5.5.4 Events Page

“Events” button on the Profile page will lead the user to the Events page. Events page contains events that user is planning to attend or the future events that created by the user. Any past events will not be shown on this page. shortly, this page is a combining of the pages “My Events” and “Future Events” to ease the user to track them.

Figure 13: Events Page

Touching the “Join Events” button, user will be directed to “search interests”page (3.5.5.7) in order to select which of the interests event he would like to join.

The back button of the Android phone will redirect user to the main page.

2.5.5.5 Find Friends Page

On the Main page, “find friends” button redirects the user to the Find Friends page which helps the user to find his friends by entering some letters or full of their names. In the search results, Users already friended will be shown in another color. touching the names in the search bar will lead the user to corresponding result s Profile page.

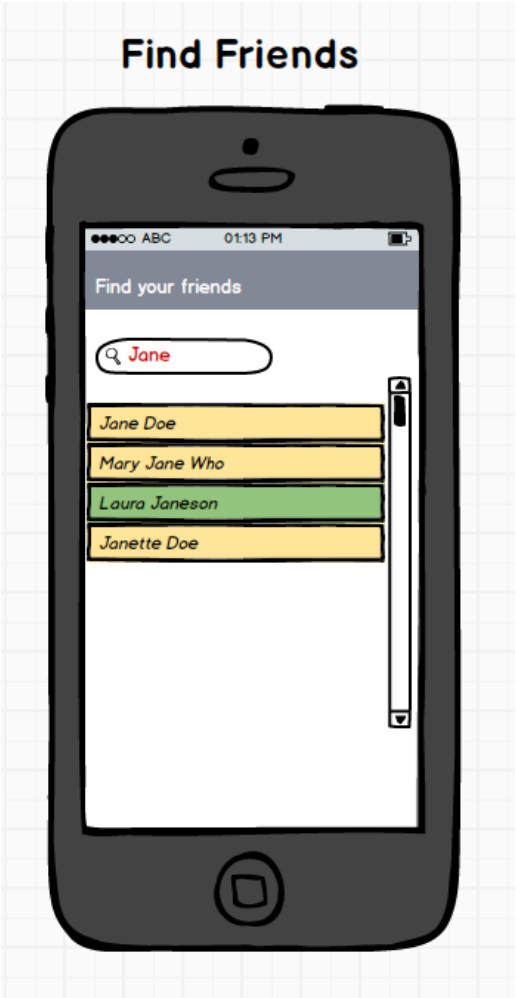

Figure 14: Search Friends

The back button of the Android phone will redirect user to the main page.

2.5.5.6 Other User's Profile Page

The Other User's Profile Page contains Buttons which are "My Events", "Interests", "Joined Events", "Future Events" and "Contact Info". Each of them will redirect the user to a corresponding pages shown below.

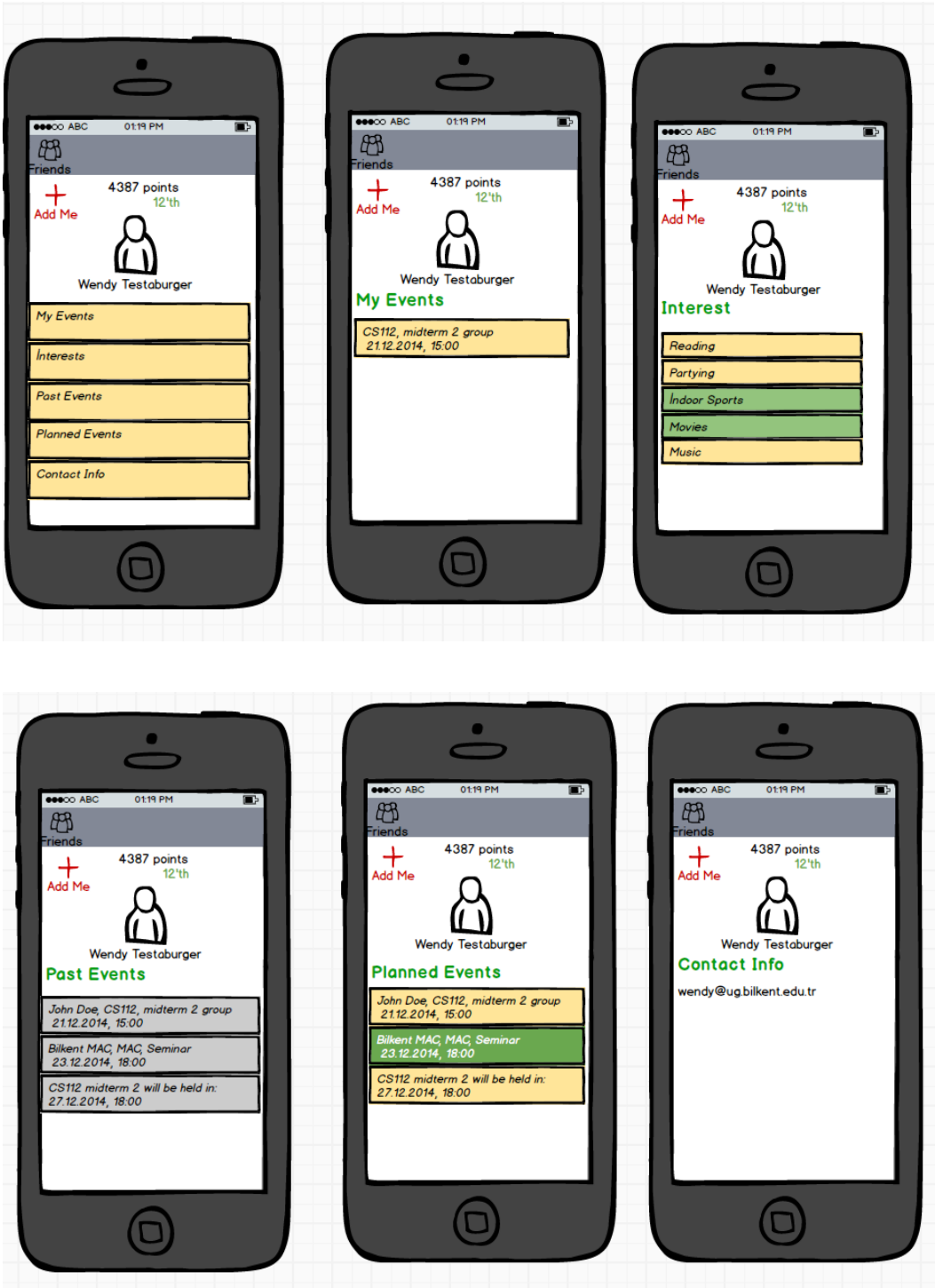

Figure 15:other user's profile

In “My Events” page, user will look up the events that the corresponding user created.

“Interests” page will show the corresponding users selected interests.

“Past Events” page contains the events that the user joined.

“Planned Events” page contains the events that the corresponding user is planned to join.

“Contact Info” contains the university-mail of the user.

By touching the “add me” buttons on the pages, user will send a request to the corresponding profile. Those request will be sent to her notifications.

From those five subpages,by touching the back button of the Android phone, user will be leaded back to the other user’s profile page. From the Profile page, user will be redirected to his own Main Page by touching the back button.

2.5.5.7 Search Interests Page

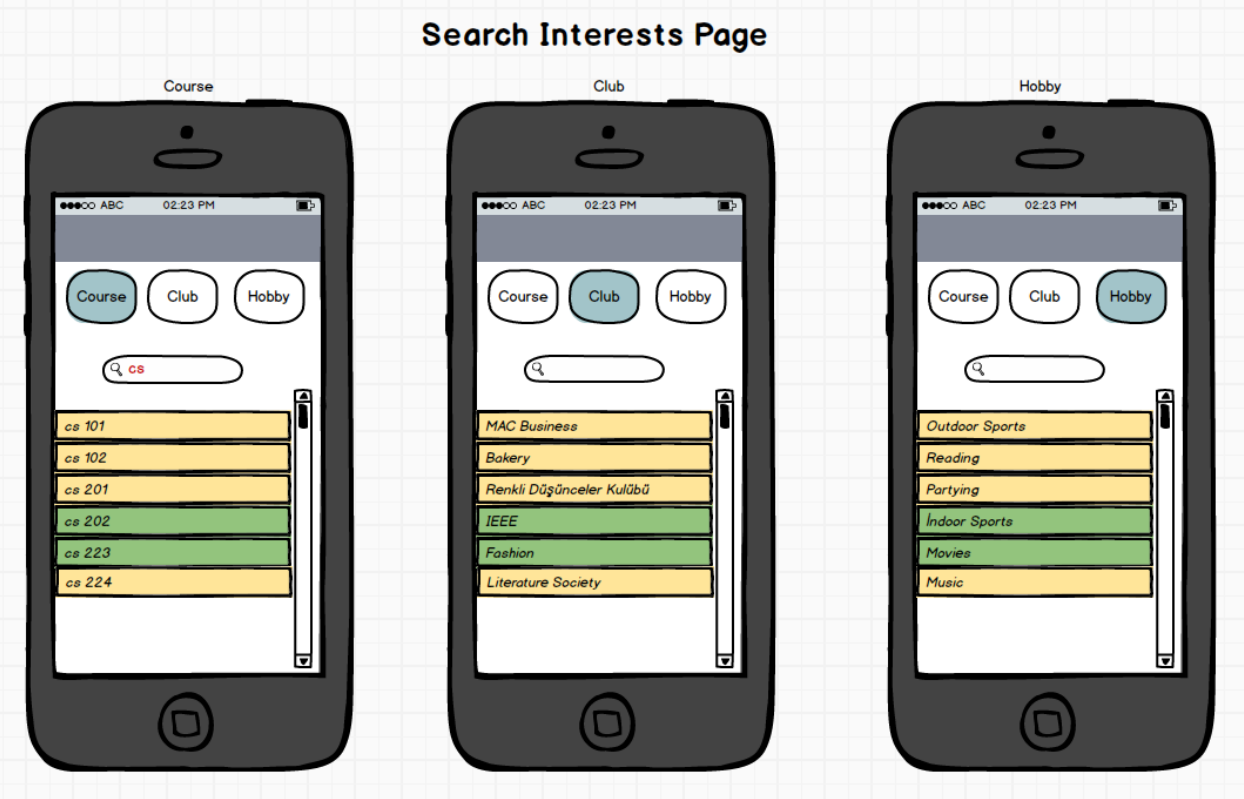

Figure 16:search interests

Select Interests page can be reached from the “interests” subpage of the profile page by touching the add interests button. This interests has three categories: “Course”, “Club” and ”Hobby”.

.Course contains the all courses given in the university. They are taken from the Stars System.

.Club contains the University Clubs which have been a member of our system. By selecting some of them, user will be able to see those club events in his news feed.

.Hobby category contains some hobbies can be performed in the campus.

From the interests, already selected ones will appear in a different color in order to ease the usage.By touching one of the interest from the list's, the detail page(3.5.5.8) of the corresponding interest will be shown.

2.5.5.8 Interest Details Page

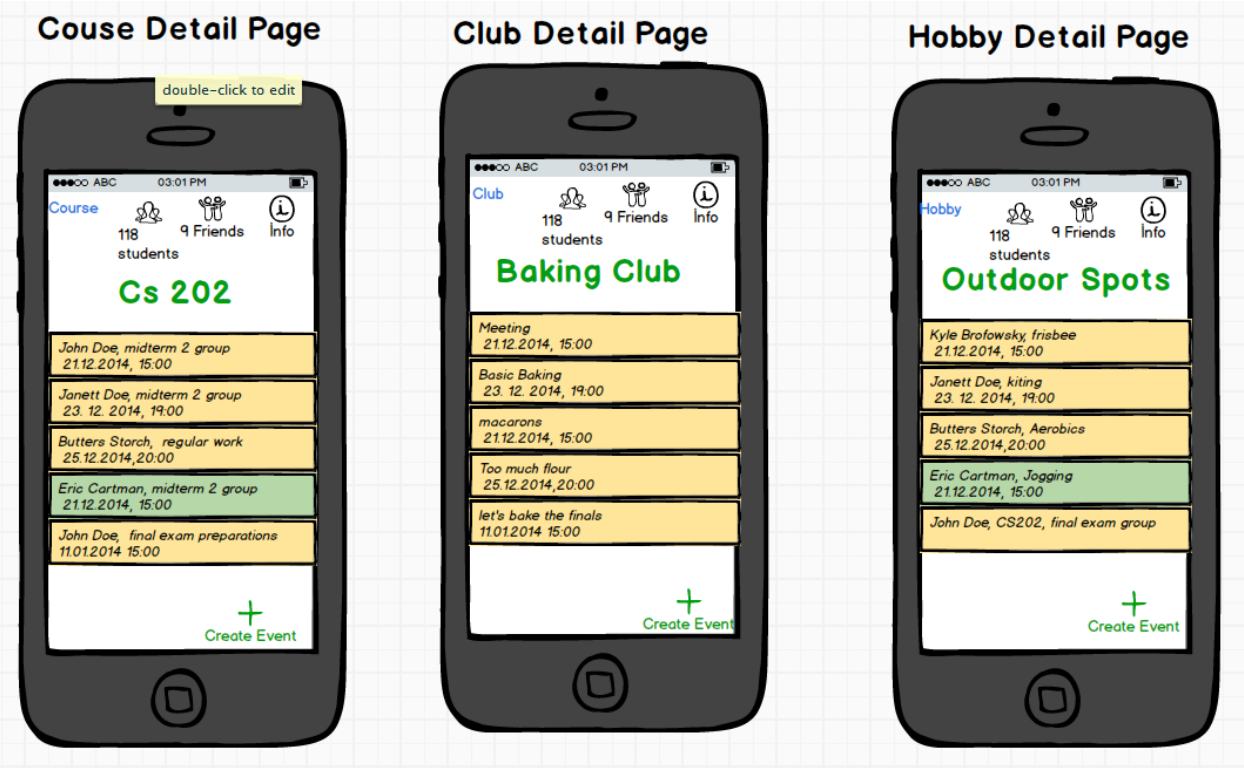

Figure 17:interests details

Interest Details Page contains the details of the corresponding interest.:

- Which category it belongs to
- How many students are interested in it

- How many of them are your friends
- İnfo text of the interest
- Future events created for this interest

The events created by the user himself will be shown in a different color.

By touching the button “Create Event“, user will be redirected to the Create Evet page.

2.5.5.8 Create Event Page

The Create Event page can only be reached by “Interest Details Page” in order to create an evet about corresponding event. To create one, user should enter a Title for his event, give some information , enter the date, time and place.

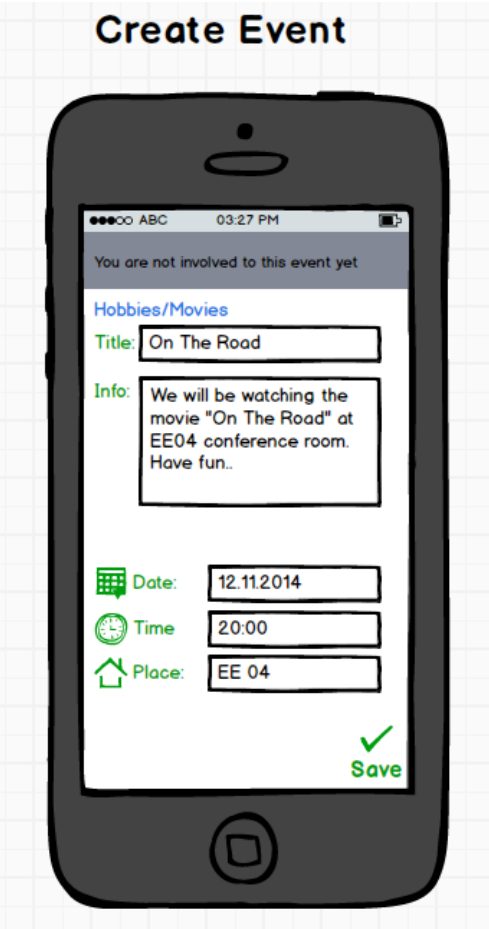

Figure 18: Create Events

By touching the save button, the Event will be added to the “Events” subpage of the user’s main page and “My Events” subpage of the user’s profile page. Then, the user will be redirected to the main page.
Touching the back button of the Android phone will redirect the user to the main page without creating an event.

2.5.5.8 Event Detail Page

When the user wishes to know about an event, he can select it from the “Interests Detail Page”. When the corresponding event is touched, It’s detail page will be shown. The page contains

- Which category event belongs to
- Name of the event
- İnfo text of the event
- Creator of the event
- How many students are planning to join the event
- Date, time and place

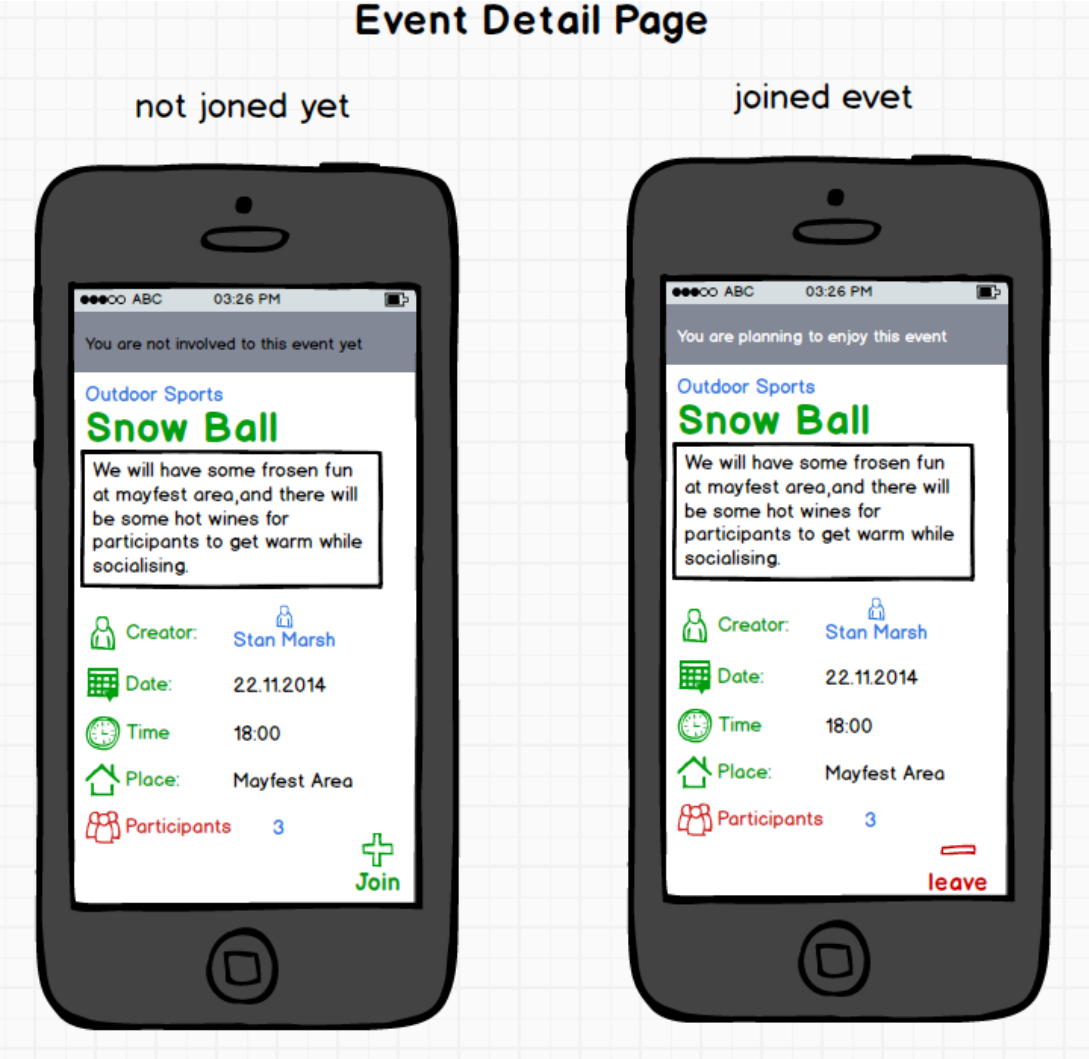

Figure 19:Event Details

If the user wishes to join the event, he may touch the “Join” button. If the event details page belongs to an already planned event’s page, there appears a leave button for user to leave the event.

If the event is created by the users self, rather than join or leave button, there will be “edit” and “cancel” buttons. When an event is cancelled or edited, a notification will be sent to the participants.

3.Glossary

Sql: Sql is Stuctural Query Language designed to manipulate relational database management systems.

HTTP: Hypertext Transfrer Protocol

Hibernate: “Hibernate is a object relational framework which enables developers to wrte applications more easily, whose data outlives the application process”[2]

NFC: Near Field Communication technology that will help us to check if the students are together.[3]

Json: Javascript object notation

4.References

1. From Microsoft: "Structured Query Language, invented at IBM in the 1970s. It is more commonly known by its acronym, SQL .." Microsoft."Structured Query Language (SQL)".

2.http://hibernate.org/orm/[2]

3. "NFC Forum Publishes Specification For "SmartPoster" Records". 5 October 2006. Retrieved 14 June 2011.

4. http://www.json.org/