



Capstone Project – Phase A

Co-Bie

An application to find people with common hobbies

Project Code: 23-1-D-15

Supervisor

Prof. Zeev Brazily

zbarzily@braude.ac.il

Raz Avraham – 314788001

Raz.Avraham@e.braude.ac.il

Amit Kempner – 206078537

Amit.Kempner@e.braude.ac.il

Table of contents

1. Abstract.....	3
2. Introduction.....	4
1. The problem and motivation.....	4
2. Stakeholders.....	4
3. Our solution.....	5
4. Organization of the paper.....	5
3. Background.....	6
1. Related Work.....	6
2. Survey.....	7
3. Agile Development.....	8
4. Mobile Development.....	8
5. Android.....	8
6. Client-Server Model.....	9
7. Mobile cloud computing.....	9
4. Expected Achievements.....	10
1. Project Goals.....	10
2. Unique Features.....	10
3. Engineering Challenges.....	11
5. Engineering Process.....	12
1. Process.....	12
2. Product.....	13
1. Use-Case Diagram.....	13
2. Class Diagram.....	14
3. Activity Diagram.....	14
1. Create Event.....	15
2. Join Event.....	17
3. Add Review.....	18
4. Software Architecture Diagram.....	19
5. User Interface.....	20
1. Login Screen.....	20
2. Registration Screen.....	21
3. Home Screen.....	22
4. Join Event Screen.....	23
5. Create Event Screen.....	24
6. Menu Screen.....	25
7. Review Screen.....	26
6. Evaluation / Verification Plan.....	27
1. Unit Testing.....	27
2. Functional Testing.....	29
7. References.....	31

Abstract

Co-Bie application is a convenient and effective way for individuals to connect with others who share similar interests. Through the use of this application, individuals can share tips and advice, and organize meetups and events to participate in their hobby together.

The document provides a comprehensive examination of a proposed application that connects individuals with shared hobbies. To establish the necessity of the project, the document explains the motivation and purpose behind the development, as well as the solutions proposed to address the challenges. In-depth research was conducted in the relevant field, including a survey to gather insights from the target audience. Based on the research, the document outlines the key features the application will offer to users.

To ensure the successful development of the application, the document delves into the engineering challenges that may arise, and how they will be addressed. The application's functionality and interactions are visualized through the use of diagrams, such as use-case, class, and activity diagrams. The document also presents the expected user interface structure and software architecture, providing a clear understanding of the application's design and functionality.

To ensure the product meets the set goals and provides a high-quality user experience, the document also outlines a plan for testing the application. Overall, the document presents a thorough and well-structured plan for the development of the application, highlighting the steps that will be taken to ensure its success.

Introduction

The problem and motivation:

Nowadays, all over the world, there are many people with different hobbies. Many of these people have no one to share their hobbies with. Our goal is to solve the problem by creating a connection between people with common hobbies. There are platforms that provide a partial solution to the problem. Those platforms make it possible to meet new people based on shared hobbies but do not focus on the possibility of setting specific events. Our solution is based on creating a meeting spontaneously and quickly.

Stakeholders:

The potential stakeholders of our system are:

- People from all the population layers who do not have partners to share their hobbies with.
- People who are looking for new partners for diversity and new friendship.
- People and organizations interested in organizing a tournament/competition in a certain field (for example, a municipality organizing a basketball tournament for the city's residents).
- People with disabilities who are unable to leave their homes and want to share their hobby virtually.

Our solution:

We offer an Android app to solve the problem. The application will include an option to create a meeting along with relevant details (purpose, number of participants, location and time). The users of the application will have the option to register for the meeting and confirm arrival. In addition, it will be possible to conduct a live chat between the participants of the meeting.

Each user of the application will have basic identifying information (name, age, area of residence, photo and interests). Moreover, there will be an option to report an untrustworthy user, these reports will be a measure of reliability.

The interface will contain two main types of events: physical and virtual.

The physical event interface will be displayed on a map. Meetings that take place in the user's area will be displayed on the map. The virtual event interface will be displayed as a list sorted by the hobbies the user has chosen.

When a new meeting that matches user's hobby is created, a notification will be sent to him with an option to join the meeting.

Organization of the paper:

- **Section 1:** Explain the problem and our proposed solution.
- **Section 2:** Describe the relevant background of the project, which includes the technological background and the architecture we will use for the project.
- **Section 3:** Define our expected achievements, our main goals and the application's unique features.
- **Section 4:** Detail the general process of the project. In addition, we present the characterization of the product.
- **Section 5:** Present the verification plan of the system, which includes the unit testing and the functionality testing.
- **Section 6:** Present the relevant references that were used in that project.

Background

Related Work:

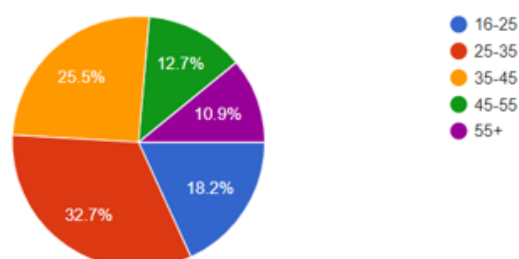
There are different platforms that offer a solution to the problem. Each platform has its own unique approach to user interaction. We will detail some of the platforms:

- **Pinterest** – Online social network similar to a bulletin board. The service allows its users to create and manage their own photo collections. The stated purpose of the site is "to connect all the people in the world through things that interest them".
The social network does not provide the option of social meeting but focuses on sharing hobbies and creating a conversation about them.
- **Meetup** – Social media platform for hosting and organizing in-person and virtual activities, gatherings, and events for people and communities of similar interests, hobbies, and professions. Meetup mainly focuses on getting to know people with similar interests but does not emphasize the existence of the hobby itself during the meeting. For example: a chess meetup focuses on getting to know people who like to play chess, but the main purpose of the meeting will not be to play the chess game.
- **Facebook** – A website operated by the American company Meta. The site is the largest online social network in the world. Facebook gathers many groups under it. There is a group for almost every topic that exists. The group is a platform where you can have conversation, get to know each other and have a joint discussion, but there is no hierarchical and organized model through which you can choose a hobby and hold a meeting.

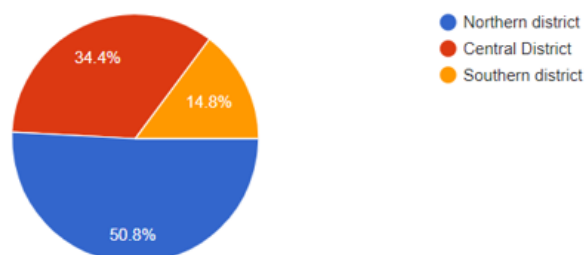
Survey:

In order to get an indication of the problem, we conducted a survey that included 73 people in a diverse age range and residential area. Below are the questions and results of the survey:

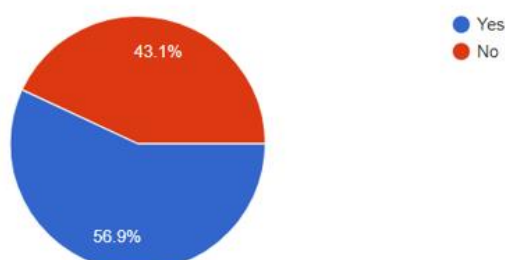
How old are you?



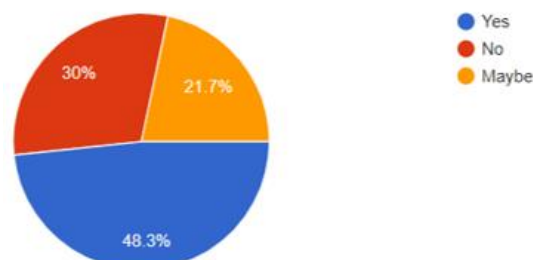
Where do you live?



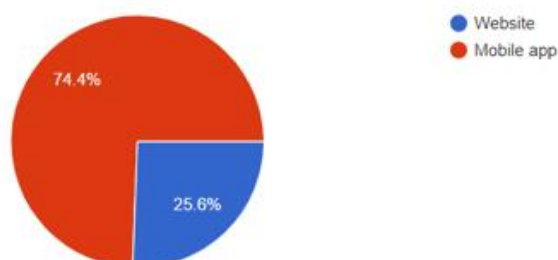
Do you have hobbies that you have no one to share them with?



Are you interested in a platform that will allow you to find partners for your hobbies?



If you are interested in the platform, would you prefer it to be on a website or mobile app?



Based on the results, we have reached several conclusions:

- The problem is prevalent among a high percentage of respondents.
- It is evident that there is a demand for a platform that will solve the problem.
- The absolute majority answered that they would prefer a mobile app platform over a website.

Agile Development:

The development methodology that will guide us in the development process of the project is Agile. Agile project can be delivered quickly and offers greater flexibility. Considering that as students we have a lot of tasks throughout the semester, we decided that this is the most suitable method for us.

Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

Mobile Development:

We chose to develop our idea as a mobile application for several reasons:

- **Ease of access** – for the user, clicking on app is far quicker and easier than accessing to website, even if they've bookmarked it.
- **Communication** – it's easier to interact with users and prospects with an app including the immediacy of being able to attract their attention and maybe prompt action through, for example, push notifications.
- **Staying "front of mind" with users** – The application icon on the mobile home screen will remind the user his need of the application.
- **GPS** – The platform is designed to be implemented on a GPS-based map. Naturally, mobile development would be more appropriate.



Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.

Android development is a form of software engineering dedicated specifically to creating applications for devices that run on the Android platform.

Client – Server Model:

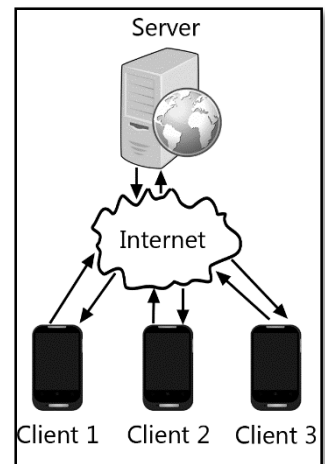
For our platform, we will require a constant connection with a server that will contains a large amount of information. For example: data about the user - identifying details, interests, etc. Therefore, we see a strong need to use a client-server model.

What is a Client – Server Model?

A client and server networking model is a model in which computers such as servers provide the network services to the other computers such as clients to perform a user based tasks. This model is known as client-server networking model.

The application programs using the client-server model should follow the given below strategies:

- An application program is known as a client program, running on the local machine that requests for a service from an application program known as a server program, running on the remote machine.
- A client program runs only when it requests for a service from the server while the server program runs all time as it does not know when its service is required.
- A server provides a service for many clients not just for a single client. Therefore, we can say that client-server follows the many-to-one relationship. Many clients can use the service of one server.



Mobile Cloud Computing:

For our platform, we will require the use of cloud services in order to get better performance and get accessible and flexible data storage. In addition, choosing a good cloud service will cover the issue of information security.



What is a Cloud Computing?

Cloud computing is the delivery of computing services such as software, databases, servers and networking, over the internet. This means end users are able to access software and applications from wherever they are.

Expected Achievements

Project Goals:

- The overarching goal of the project is to develop a friendly and easy-to-use application that will allow users to schedule meetings based on common hobbies.
- The application will allow the user to register with identification details and add personal interests under categories.
- The system will allow people with common hobbies to create social events that will help them develop their hobbies and enrich their social circle.
- The system will provide a convenient infrastructure for virtual meetings on the relevant platforms.
- The interface of the physical meetings in the application, will be based on a map. Each user will be able to create an event that includes the purpose of the meeting, time and location (which will be displayed on the map).

Unique Features:

- **Notifications** – When an event is created, notifications will be pushed for users whose interests match the event's purpose and their geographic location (if physics event) is relevant to the event.
- **Navigation** – The user will have the option to navigate to the location of the physical event using the various navigation apps available on his cell phone.
- **Live Chat** – Users registered for the event will be able to chat via live chat.
- **User Status** – The application will display a status for each user. This status is a measure of his level of activity at events. For example: "Veteran", "New", etc.
The status will be updated according to the number of events the user has participated in. One of the main purposes of the status is to be a reference for the user's reliability.
- **Report** – The application will offer the possibility to report abusive users who violated ethical rules defined during the registration process.
- **Virtual Events** – There will be an option to schedule events for hobbies that happening virtually and do not require a physical meeting. For example: online games, zoom meetings, etc.

- **Tournament management** – Unlike other features, this is a feature that is used during the event. The user will have the option to manage a tournament through the application. In this way, users can add another experiential dimension to the meeting. Of course, this feature will be relevant to competitive hobbies only.
- **Adding new friends** – the app will allow the user to add friends. Depending on the list of friends, the application will offer the user relevant meetings.
- **Event Summary** – After the event, it will be possible to add comments and intellectual achievements from the meeting.

Engineering Challenges:

- **Location management** – In order to answer the requirements of certain features in the application, we must map each user to a relevant area. The challenge we will face will be to find an efficient way to map the users while using the GPS function. The development will include a flexible algorithm as there may be real time changes.
- **Identifying Unreliable Users** – The application is based on physical encounters between people. It is our responsibility to develop an algorithm that will monitor user's safety and keep untrustworthy users away.
The system will include an algorithm based on smart use of user reports and user status in order to alert the administrator about unreliable users.
- **Matching suggested interests** – when the user registers for the application, he will choose his interests. During the use of the application we would like to manage an algorithm through which we will be able to offer him new hobbies according to various variables. For example: location, mutual friends, similarity to existing interests, etc.

Engineering Process

Process:

In order to develop the application in the optimal way, we created an orderly work process that includes several milestones:

- **Problem Analysis** – First, we explicitly defined the problem we are coming to solve. Analyzing the problem helped us understand what the root of the problem is, where it exists and how to approach it.
- **Target Audience** – By conducting surveys relevant to the problem, we were able to get a significant indication of who the customers or target audience of the application are. In addition, we understood in an unambiguous way that the problem is among different population groups and there is a need for a solution.
- **System Requirements** – When we had a full understanding of the problem and its origin, we detailed the characteristics of the system and what it would be able to perform and deliver. That is, the functional and non-functional requirements of the system based on the information we have gathered.
- **Offering Solutions** – Several potential ideas for solving the problem have come up. For each proposed idea we analyzed its advantages and disadvantages based on pre-defined key criteria (according to the surveys and research we conducted in previous stages).
- **Selected Solution** – We chose a map-based Android application. After examining the findings, we came to the conclusion that this is the optimal solution that will give the best answer to the requirements we defined.
- **Development Tools** – After finding the appropriate solution, we chose the set of tools that is suitable for the solution development. The development environment will be Android Studio while the writing of the code will be done in JAVA language combining design elements in XML.
The work with cloud services will be through the Firebase platform that offers real-time database.
- **Development Methodology** – We chose the Agile development methodology due to the flexibility it offers. Also, the way tasks are divided into sub-tasks can help and lead to efficient and fast development.

Product:

Use Case Diagram:

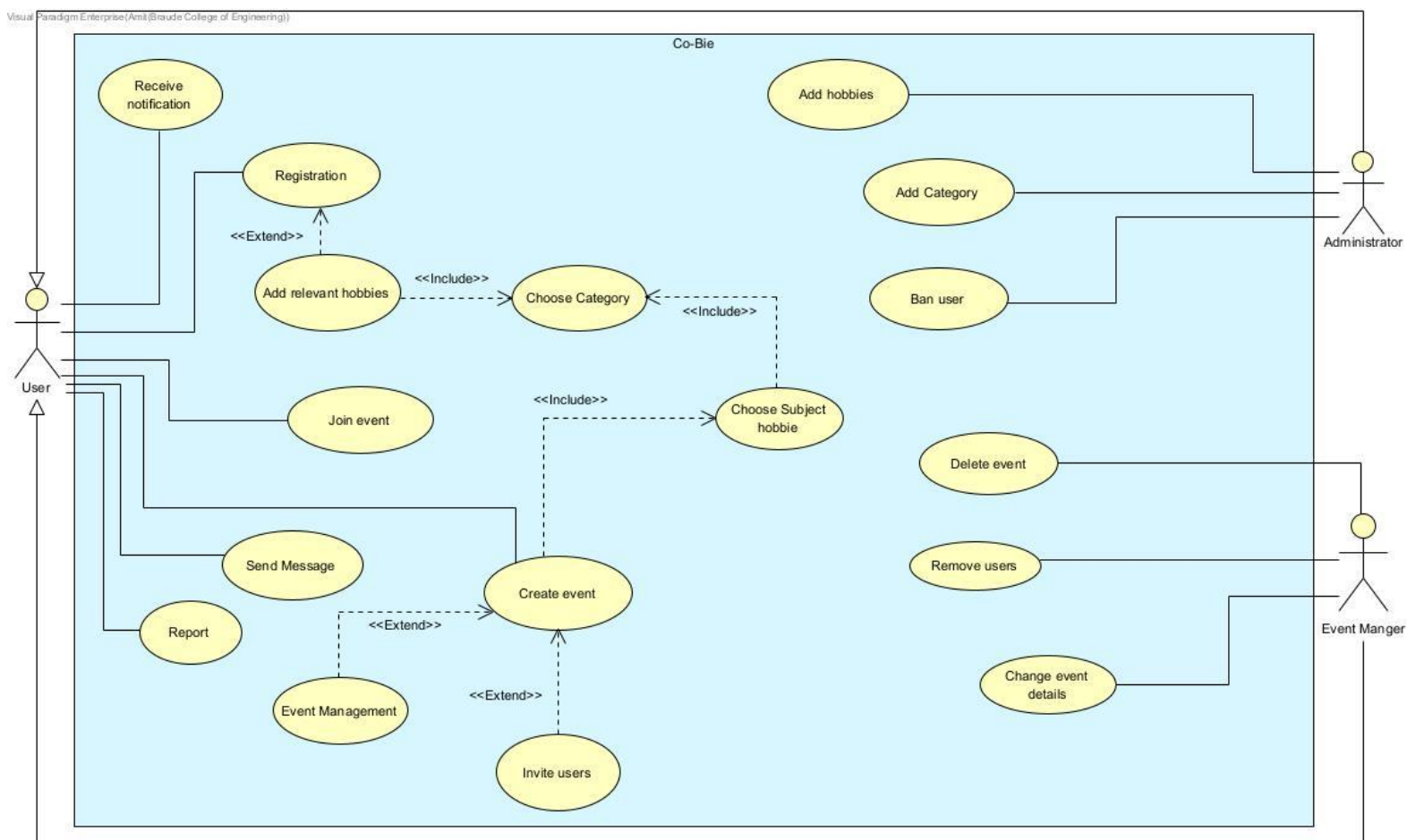


Fig 1: Use Case Diagram

Class Diagram:

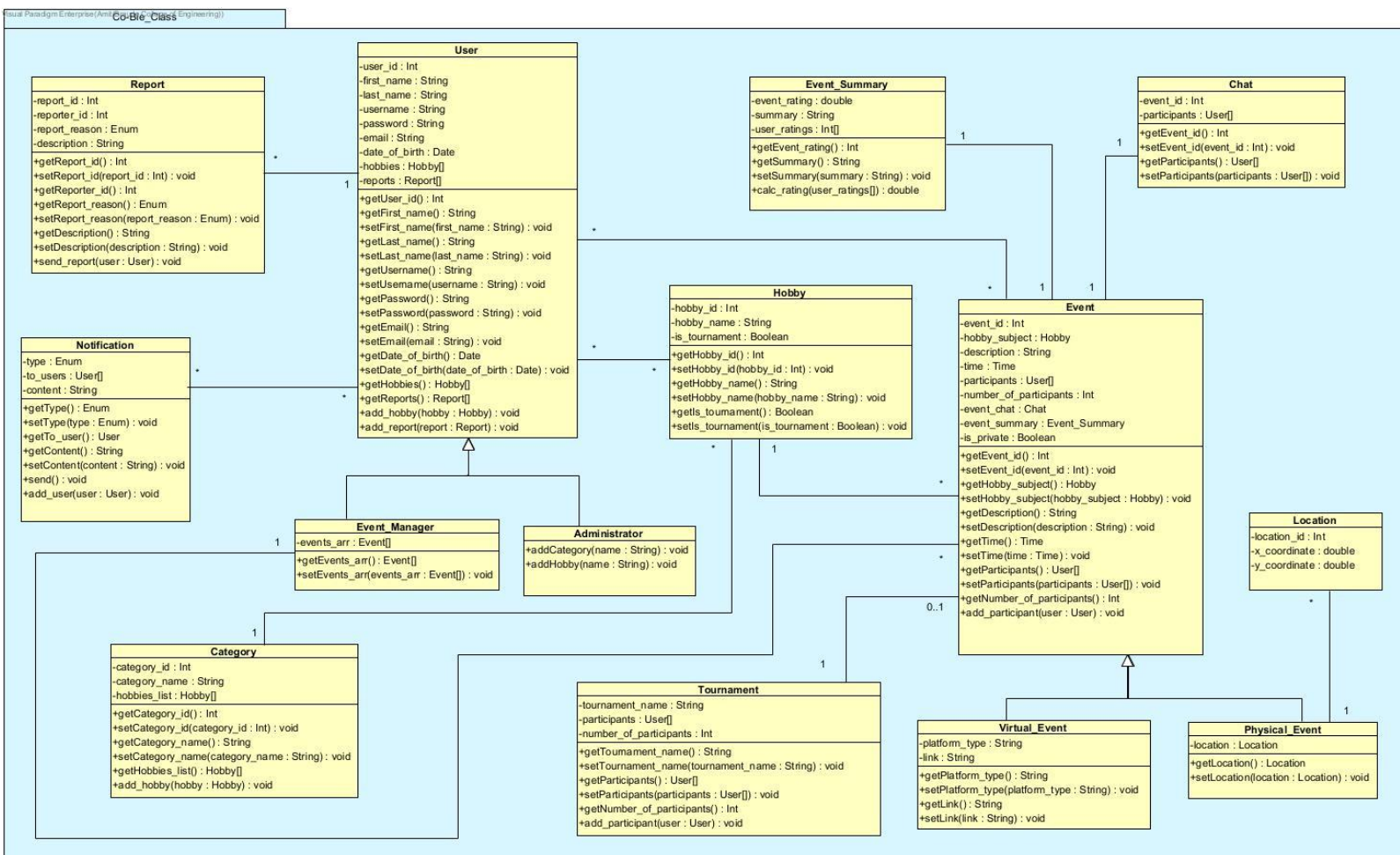


Fig 2: Class Diagram

Activity Diagram:

Create event

1. The user login to the system.
2. The system displays the home screen.
3. The user selects the menu option.
4. The system displays the menu options.
5. The user chooses "Create Event" option.
6. The system displays the event details configuration screen.
7. The user chooses the event category.
8. The system displays all the hobbies that are under the selected category.
9. The user chooses the event hobby.
10. The user chooses the event time, description, public/private, physical/virtual.
11. If the user has selected a public event, pass to step 14.
12. If the user has selected a private event, the system will pop up a window to select invited users.
13. The user chooses the invited users.
14. If the user has selected a virtual event, pass to step 17.
15. If the user has selected a physical event, the system will pop up a window to select a location.
16. The user chooses the event location, pass to step 19.
17. The system will pop up a window to insert a link.
18. The user will enter a link to the meeting.
19. The user selects "Submit".
20. The system creates and display the event.

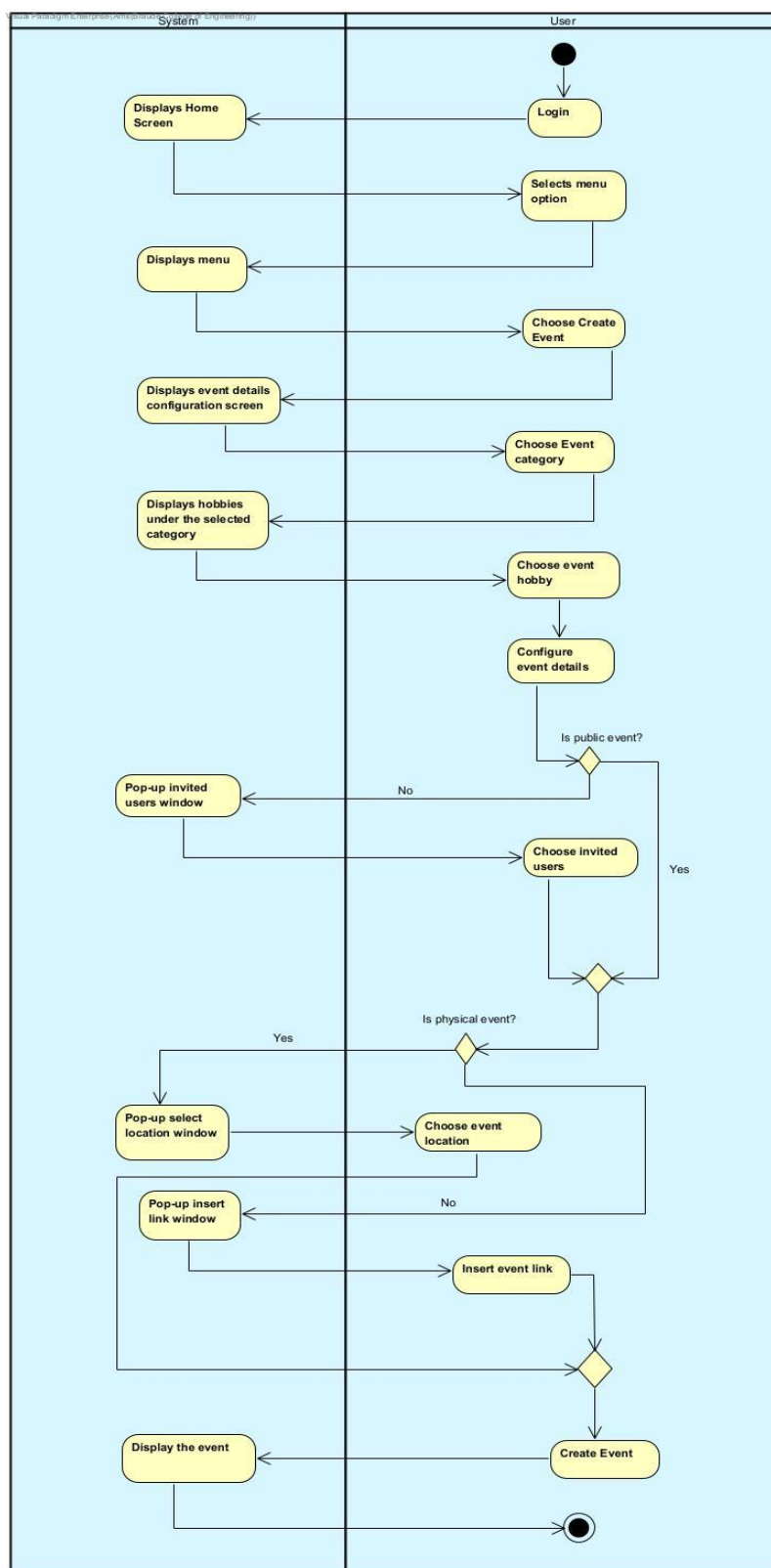


Fig 3: Activity Diagram – Create Event

Join event

1. The user login to the system.
2. The system displays the home screen.
3. The user selects “Virtual” / “Physical” tab.
4. If the customer selects “Virtual”, pass to step 6.
5. If the user selects “Physical”, The system displays a map with all existing physical public events.
6. If the user selects “Virtual”, The system displays a list with all existing virtual public events.
7. The user selects the event he wants to join.
8. The system will display the event details.
9. The user selects “Submit”.
10. The system pops up a message "You have successfully joined".

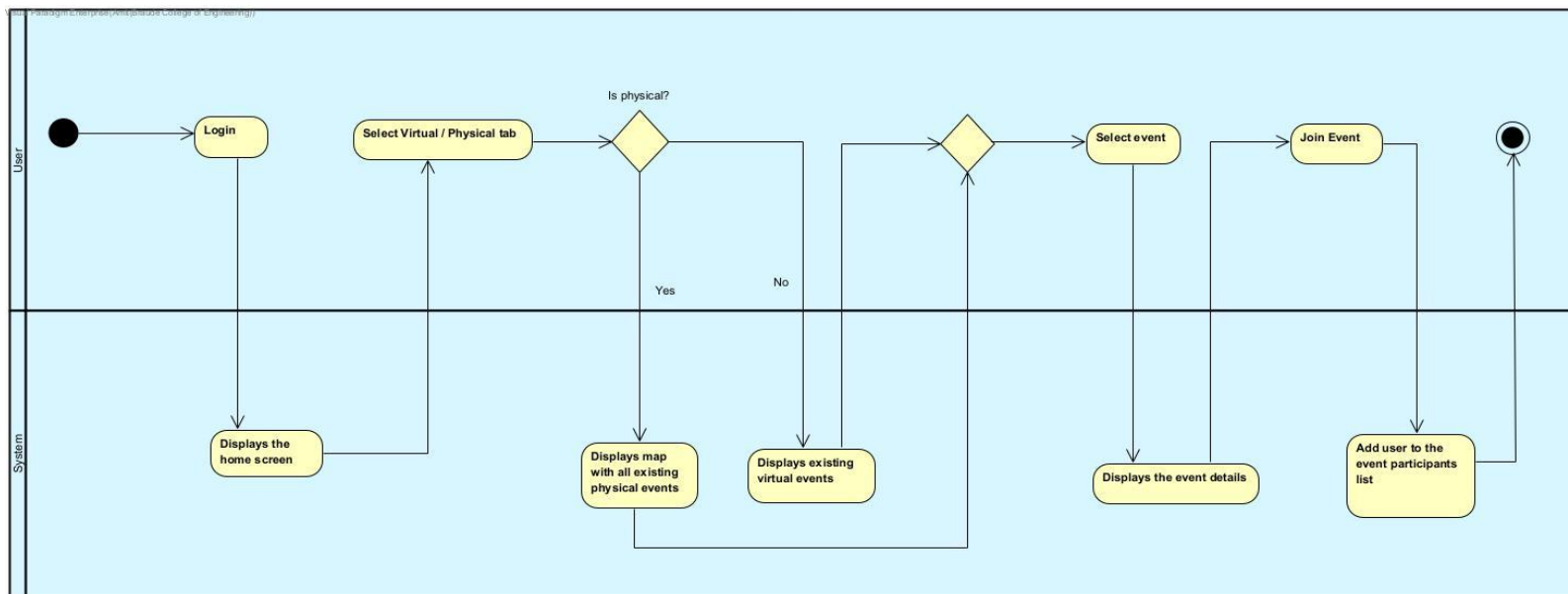


Fig 4: Activity Diagram – Join Event

Adding responses and intellectual achievements to event

1. The user login to the system.
2. The system displays the home screen.
3. The user selects the menu option.
4. The system displays the menu options.
5. The user chooses "My events" options.
6. The system displays all events that the user has joined.
7. The user selects the requested event.
8. If the event has ended, the system will display the event's comments page.
9. The user selects "Add comment".
10. The system pops up a text box.
11. The user enters his response in the text box provided.
12. The system adds the response to the response list of the event.

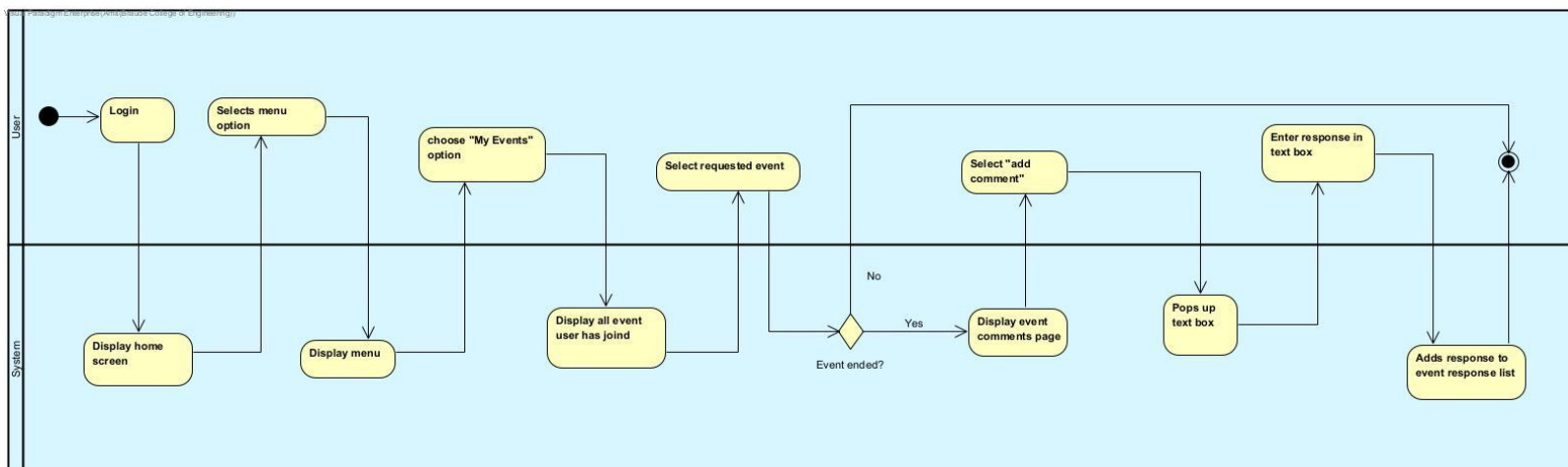


Fig 5: Activity Diagram – Add Review

Software Architecture Diagram:

Application

Services

Data Base

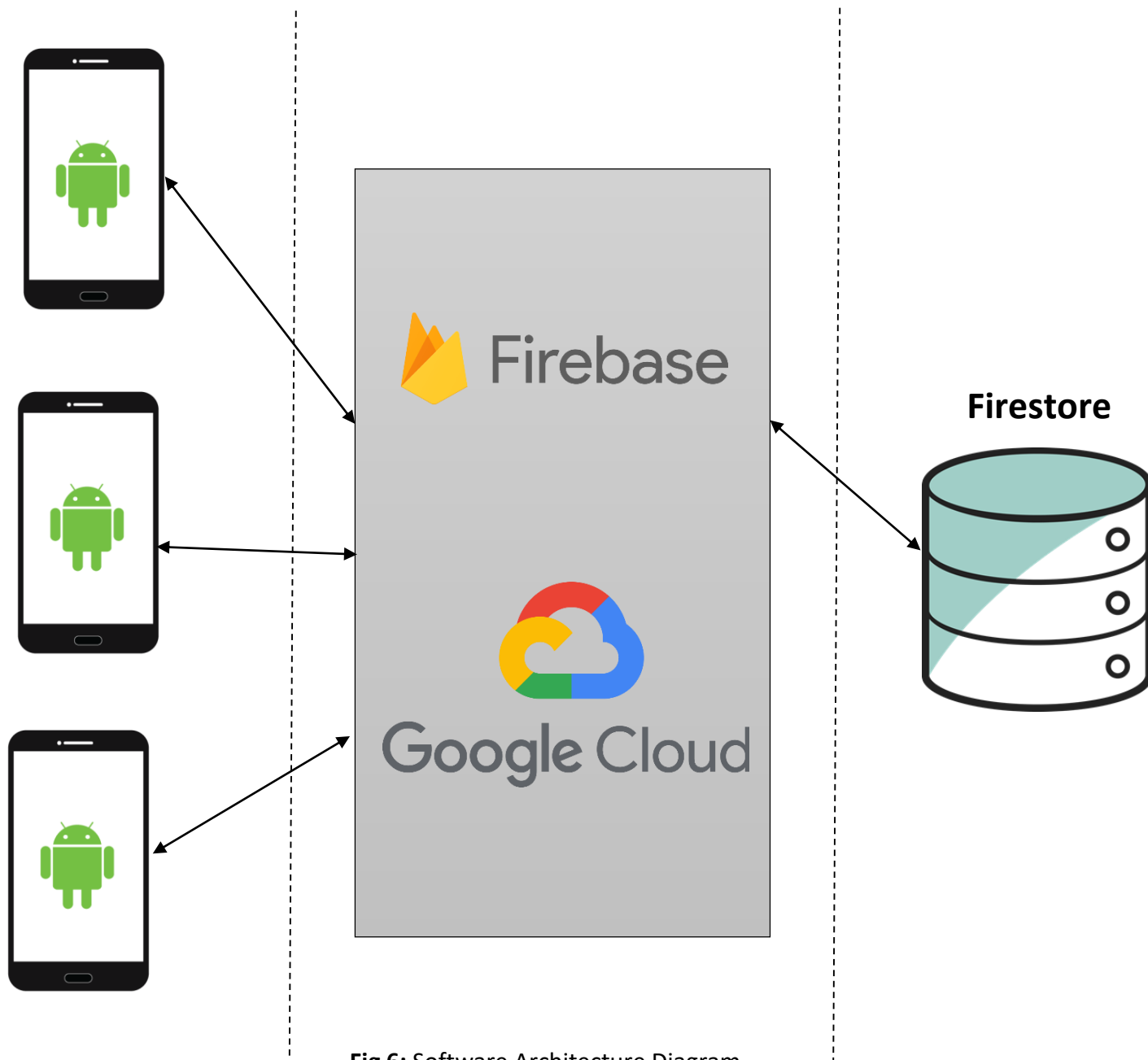


Fig 6: Software Architecture Diagram

User Interface:

- Login Screen

Login screen for registered users. For unregistered users there is an option to sign-up.

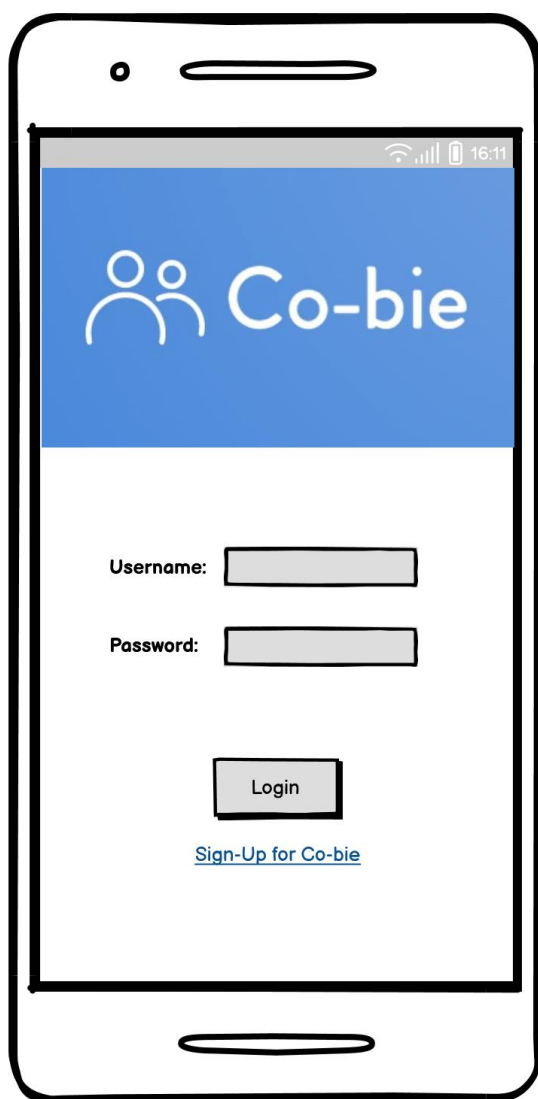


Fig 7: Login Screen

- Registration Screen

When a user clicks on "Sign-Up", he will have to enter identifying details, username and password for future login.

After entering the details successfully, he will have to select his hobbies from different categories (selection of hobbies is optional).

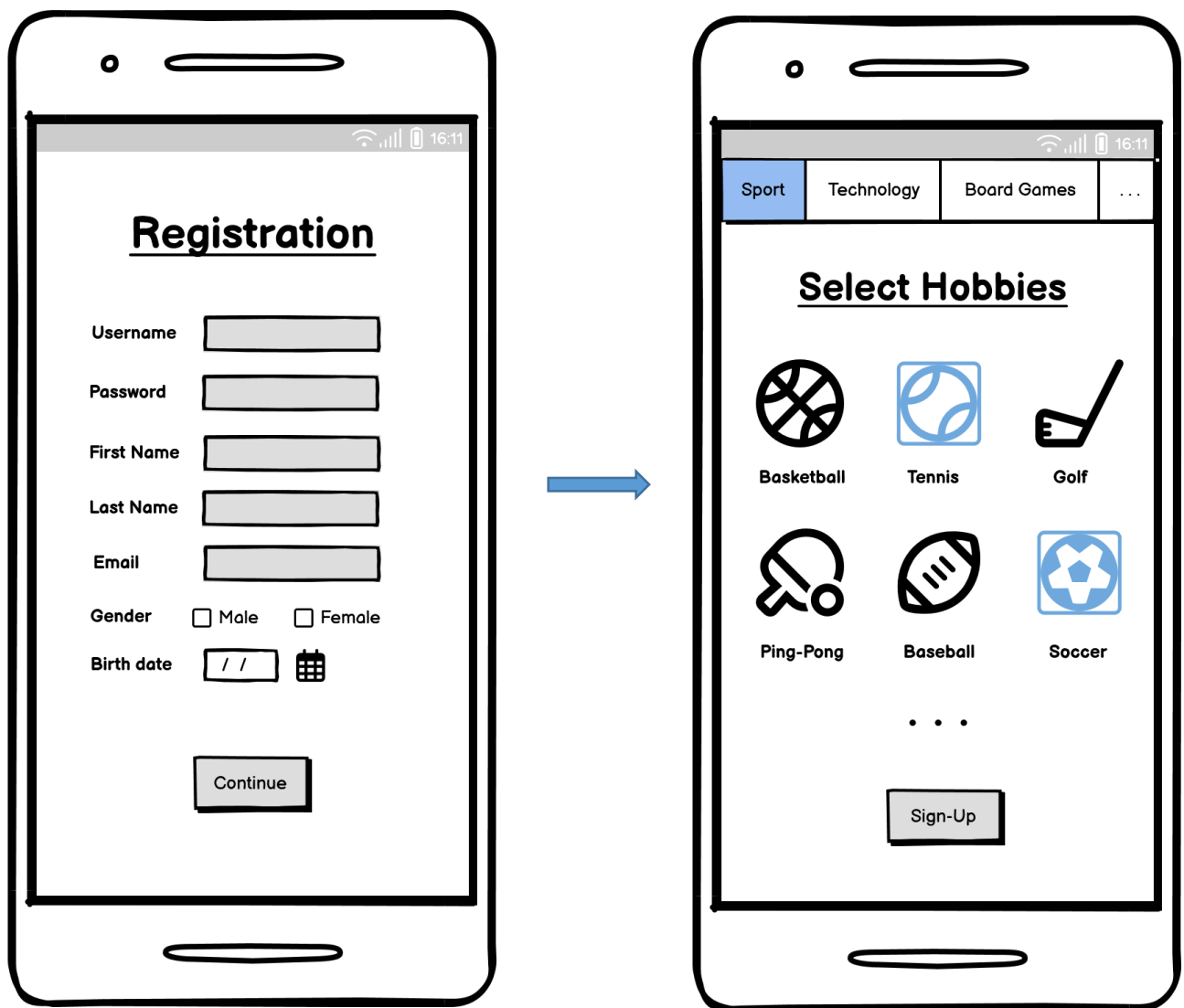


Fig 8: Registration Screens

- Home Screen:

The home page will contain the various events that exist. There will be a division into physical events and virtual events. The user will be able to click on the event he is interested in. The physical events will appear as locations on a map and the virtual events will appear in a list.

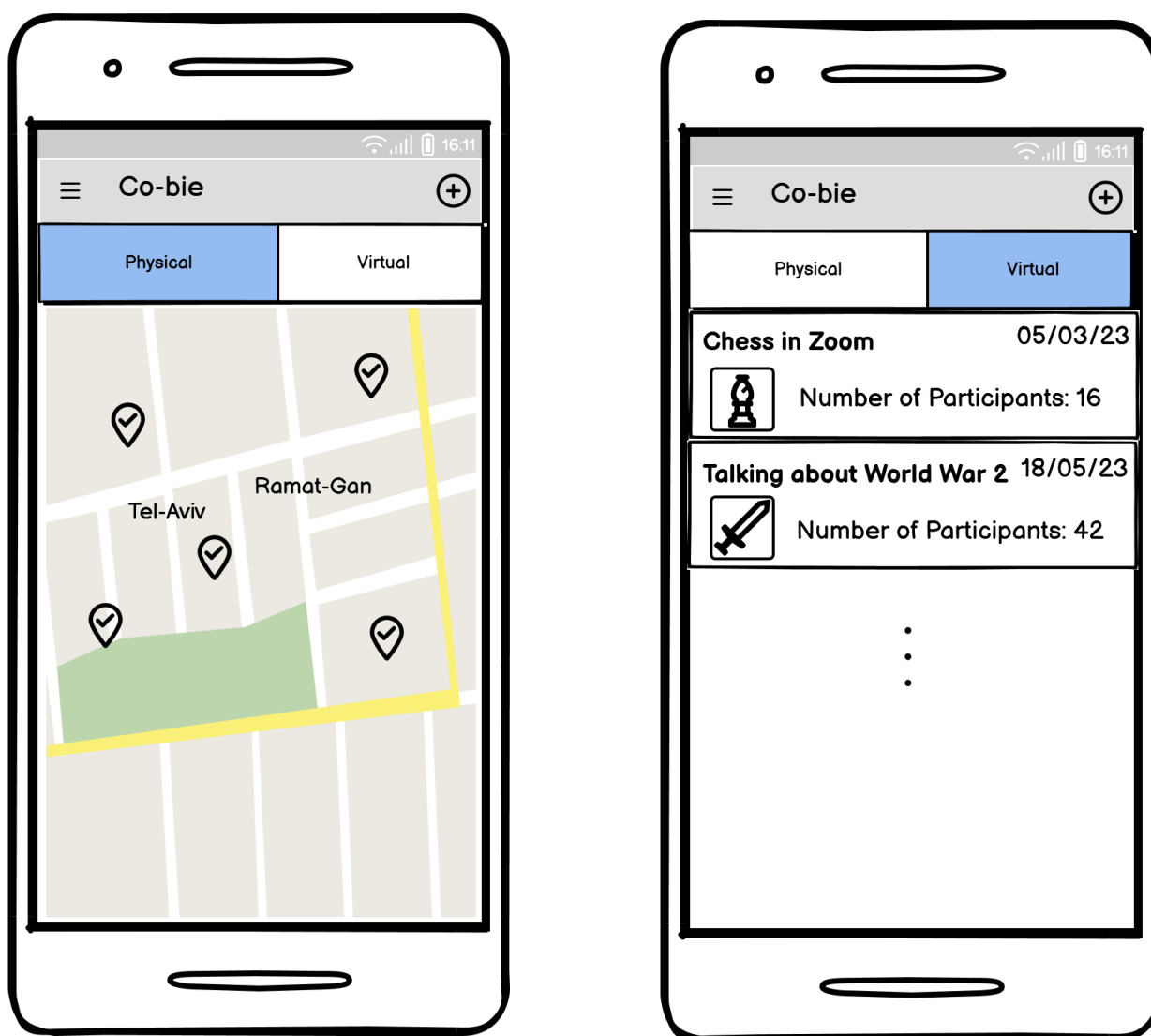


Fig 9: User Home Screens

- Join Event Screen:

When a user clicks on a certain event, a screen will be displayed with the event details and the user will be able to join the event by clicking "Join" button.

If this is a physical event, the location will be displayed.

If this is a virtual event, a link to the meeting will be attached.

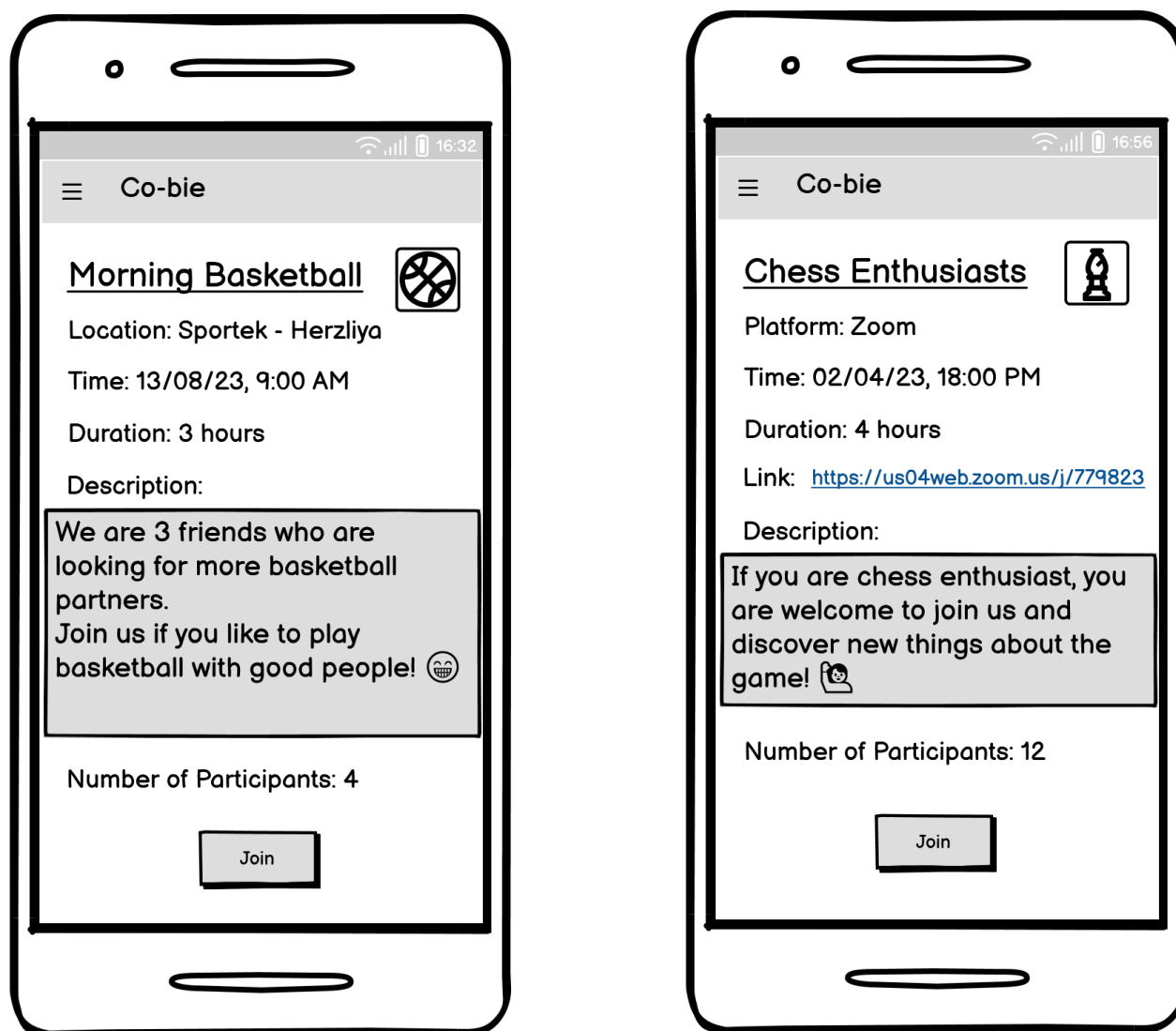


Fig 10: Join Event Screens

- Create Event Screen:

When clicking the plus button on the home screen (top right) the user will have the option to create an event.

The user will be required to enter the details of the event.

If the event is physical, the user will be asked to provide a location on a map.

If the event is virtual, the user will be required to provide the name of the platform and a link to the meeting.

If the event is private, the user will require to provide the invited users.



The image shows a mobile application interface for creating an event. The screen is titled "Co-bie" and has a status bar at the top showing the time as 17:09. The main heading is "Configure your Event Deatils". Below this, there are several input fields and checkboxes:

- Select Hobby:** A dropdown menu showing "Ping-Pong".
- Date:** A date picker showing "15/05/23".
- Time:** A time picker showing "13:30 PM".
- Duration:** A text input field showing "3.5 hours".
- Title:** A text input field showing "Ping-Pong Tournament".
- Description:** A text area containing "Join us to a great Ping-Pong Tournament!".
- Public/Private:** Two checkboxes, with "Public" selected.
- Physical/Virtual:** Two checkboxes, with "Physical" selected.
- Select Location:** A button to open a location selection interface.
- Create:** A button to submit the event details.

Fig 11: Create Event Screen

- Menu Screen:

On the home screen, the user will have the option to open the application menu (top left). The menu will contain the following options:

My Profile – option to enter the user's personal page, edit details and more.

My Events – option for the user to see the events he registered for. Also, events that have already passed will be displayed in order to watch/add reviews and intellectual achievements.

Logout – disconnection from the existing user.

Information logo - information about the use of the application.

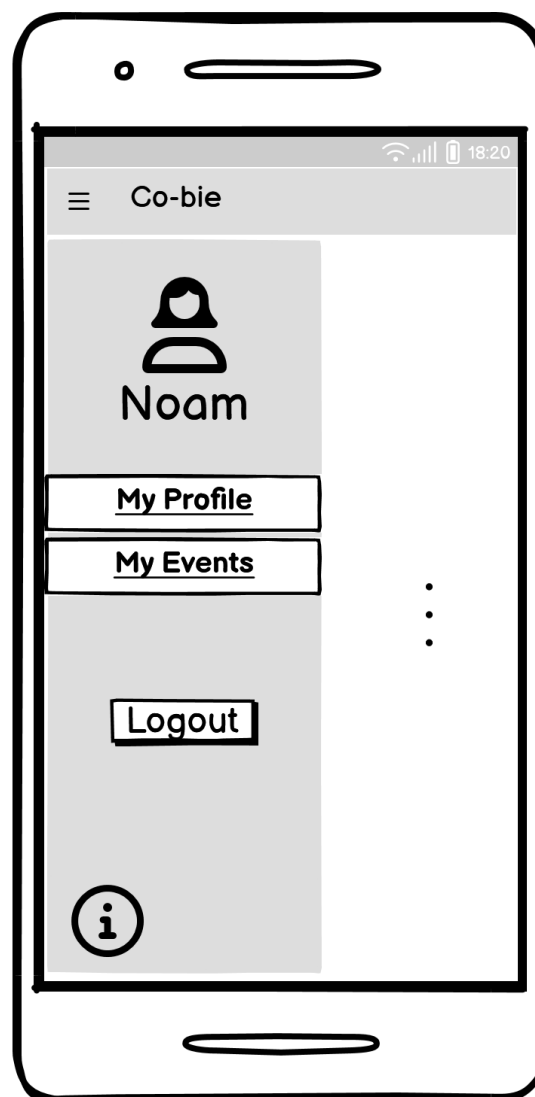


Fig 12: Menu Screen

- Reviews Screen:

When clicking on an event that has already taken place (on the My Events page), the reviews and intellectual achievements page of that event will be displayed.

The user will have the option to view reviews and also add a review.

When creating a review, the user will be required to provide a descriptive text and a rating of the event.

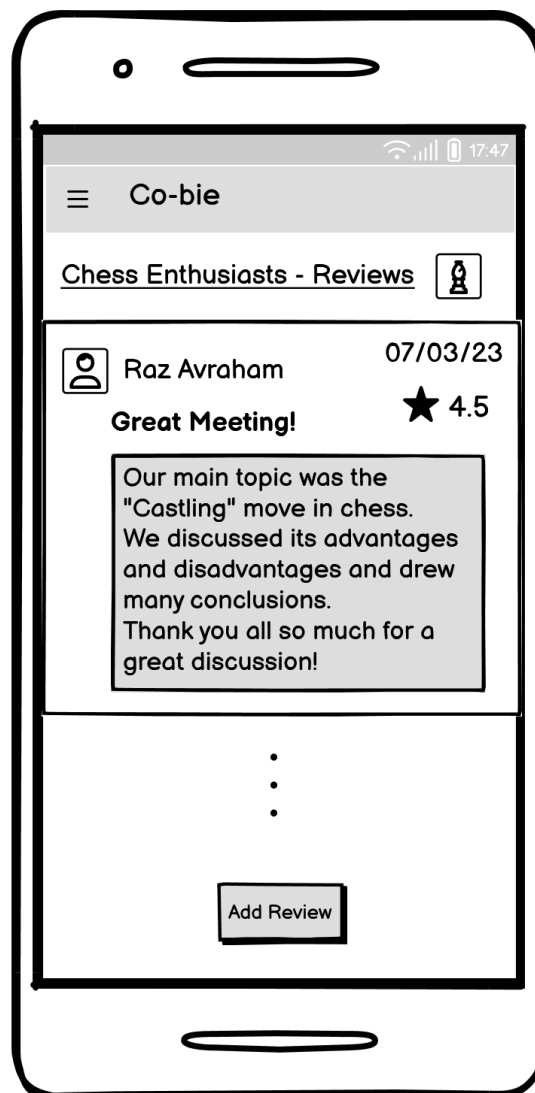


Fig 13: Add Review Screen

Evaluation / Verification Plan

In order to check that our system is working properly, we will perform 2 types of tests:

- Unit Testing – A unit test is a way of testing a unit - the smallest piece of code that can be logically isolated in a system.
- Functional Testing – Functional testing is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user's expectations.

We will perform the 2 types of tests in the Android Studio environment. For the unit testing we will use the Junit 5 framework and for the functional testing we will use the Espresso testing framework.

Unit Testing

Registration		
Number	Test Subject	Expected Result
1	Registration succeeded	User added to the users table in the DB with appropriate details
2	Registration failed (as a result of invalid fields)	User has not been added to the users table in the DB
3	Registration failed (as a result of system failure)	User has not been added to the users table in the DB - throw an exception

Login		
Number	Test Subject	Expected Result
1	Login succeeded	Username and Password exists in the DB
2	Login failed (as a result of invalid fields)	Username and Password doesn't exist in the DB
3	Login failed (as a result of system failure)	throw an exception

Create an Event		
Number	Test Subject	Expected Result
1	Event created successfully	Event added to the event table in the DB with appropriate details
2	Event creation failed (as a result of invalid fields)	Event has not been added to the events table in the DB
3	Event creation failed (as a result of system failure)	Event has not been added to the events table in the DB - throw an exception

Join to Event		
Number	Test Subject	Expected Result
1	The user joined the event successfully	User has been added to the event users list and the number of participants increased by one
2	Joining failed (as a result of system failure)	User has not been added to the event users list – throw an exception

Menu		
Number	Test Subject	Expected Result
1	Select Logout	User status have been changed to false in the DB
2	Select My Profile and edit details	Users details have been updated in the DB

Add Review		
Number	Test Subject	Expected Result
1	The user add review successfully	Review has been added to the event reviews list in the DB
2	Failed to add review (as a result of invalid fields)	Review has not been added to the event reviews list in the DB
3	Failed to add review (as a result of system failure)	Review has not been added to the event reviews list in the DB – throw an exception

Add Hobby (as Administrator)		
Number	Test Subject	Expected Result
1	The administrator adds new category successfully	Category has been added to the Categories table in the DB
2	The administrator adds new hobby successfully	Hobby has been added to the Hobbies table in the DB
3	Failed to add new category (as a result of invalid fields)	Category has not been added to the Categories table in the DB
4	Failed to add new category (as a result of system failure)	Category has not been added to the Categories table in the DB – throw an exception
5	Failed to add new hobby (as a result of invalid fields)	Hobby has not been added to the Hobbies table in the DB
6	Failed to add new hobby (as a result of system failure)	Hobby has not been added to the Hobbies table in the DB – throw an exception

Functional Testing

Registration		
Number	Test Subject	Expected Result
1	Enter a username that already exists in the system	System Displays: "The username already exists, try another"
2	Enter a password shorter than 8 characters	System Displays: "The password is too short, please enter at least 8 characters"
3	Enter an email that already exists in the system	System Displays: "The email address is already being used"
4	Enter invalid email address	System Displays: "Invalid email address"
5	According to the date of birth entered, the user is under the age of 16	System Displays: "The use of the application is only allowed from the age of 16 and above"
6	There is an empty field	System Displays: "Please fill all required fields"
7	The registration has been successfully completed	System switches to user home page screen

Login		
Number	Test Subject	Expected Result
1	There is an empty field	System Displays: "Please fill all required fields"
2	Enter a password shorter than 8 characters	System Displays: "The password invalid"
3	Enter a username that doesn't exist in DB	System Displays: "Username or Password is incorrect"
4	Enter an incorrect password	System Displays: "Username or Password is incorrect"
5	Successful login	System switches to user home page screen

Create an Event		
Number	Test Subject	Expected Result
1	Invalid time	System Displays: "Please enter valid time"
2	There is an empty field	System Displays: "Please fill all required fields"
3	No hobby selected	System Displays: "Please select the event hobby"
4	If the event is physical and no location selected	System Displays: "Please select the event location"
5	If the event is virtual and no link has been inserted	System Displays: "Please enter the event meeting link"
6	User selected physical event	System displays a map to choose a location to the meeting
7	User selected virtual event	System displays a text box to insert a link to the meeting
8	Event created successfully	System switches to the event screen

Add Review		
Number	Test Subject	Expected Result
1	The review field is empty	System Displays: "Please fill the review"
2	No rating selected	System Displays: "Please rate the meeting"
3	The review added successfully	System switches to the event reviews screen with the added review

Join to Event		
Number	Test Subject	Expected Result
1	The user joined the event successfully	System Displays: "Joined Successfully" System switches the "Join" button to "Unjoin" button

Menu		
Number	Test Subject	Expected Result
1	Select Logout	System switches to the login screen
2	Select My Profile	System switches to the user profile screen
3	Select My Events	System switches to the user events screen

References

Related Work:

- Pinterest – <https://en.wikipedia.org/wiki/Pinterest>
- Meetup – <https://en.wikipedia.org/wiki/Meetup>
- Facebook – <https://en.wikipedia.org/wiki/Facebook>

Survey:

- Google Forms - <https://www.google.com/forms/about/>

Agile Development:

- Definition – <https://www.javatpoint.com/software-engineering-agile-model>

Android:

- Definition – [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))

Client-Server Model:

- Definition – <https://www.javatpoint.com/computer-network-client-and-server-model>

Mobile Cloud Computing:

- Definition – <https://azure.microsoft.com/en-in/resources/cloud-computing-dictionary/what-is-cloud-computing/#benefits>

Espresso Testing Framework:

- Definition – <https://developer.android.com/training/testing/espresso>

JUnit Testing Framework:

- Definition – <https://en.wikipedia.org/wiki/JUnit>