#### AndroCom – Communication without Internet

#### Umer Ahmed BCS203182

#### Muhammad Harris BCS203193

#### Wasia BCS203233

#### Reset password | CUST

**Spring - 2024**

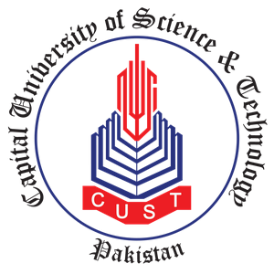
**Supervised By**

**Mr. Bilal Ahmed**

**Department of Computer Science**

**Capital University of Science & Technology, Islamabad**

|  |  |  |
| --- | --- | --- |
|  | Submission Form for Final-Year  PROJECT REPORT |  |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Version** | | V 3 | |  | **No. of Members** | 3 |
|  | | | | | | |
| **Title** | AndroCom – Communication without Internet | | | | | |
|  | | | | | | |
| **Supervisor Name** | | | Mr. Bilal Ahmed | | | |

|  |  |  |
| --- | --- | --- |
| **Member Name** | **Reg. No.** | **Email Address** |
| Umer Ahmed | BCS203182 | [Umerahmed1000@gmail.com](mailto:Umerahmed1000@gmail.com) |
| Muhammad Harris | BCS203193 | [Harris20014@gmail.com](mailto:Harris20014@gmail.com) |
| Wasia | BCS203233 | [Wasiaibrar9892@gmail.com](mailto:Wasiaibrar9892@gmail.com) |

|  |  |  |
| --- | --- | --- |
| **Member’s Signatures** |  | |
|  |  | **Supervisor’s Signature** |
|  |  |
|  |  |

**APPROVAL CERTIFICATE**

This project, entitled as “AndroCom – Communication without Internet” has been approved for the award of

**Bachelors of Science in Computer Science**

**Committee Signatures:**

Supervisor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Mr. Bilal Ahmed)

Project Coordinator: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Mr. Bilal Ahmed)

Head of Department: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Dr. Abdul Basit)

**DECLARATION**

*I/We, hereby, declare that “No portion of the work referred to, in this project has been submitted in support of an application for another degree or qualification of this or any other university/institute or other institution of learning”. It is further declared that this undergraduate project, neither as a whole nor as a part thereof has been copied out from any sources, wherever references have been provided.*

|  |
| --- |
| **Member’s Signatures** |
|  |
|  |
|  |

**ACKNOWLEDGEMENTS**

*We are deeply grateful to Allah for helping us to successfully complete this project. Furthermore, we extend our heartfelt appreciation to our parents for their unwavering support and valuable guidance, which were crucial in various phases of the project. We would also like to express our sincere gratitude to our supervisor, Mr. Bilal Ahmed, for his invaluable guidance and support throughout this project. His insightful suggestions and instructions were pivotal to our success.*

**Executive Summary**

In today's world, communication is heavily dependent on a functioning internet. If the internet were to go down, our ability to connect with each other would be severely hampered.

AndroCom offers a solution to this problem. It's an Android application designed to allow users to send encrypted text messages, make voice calls, and even video calls, all without relying on any existing network infrastructure. AndroCom achieves this by creating a localized ad hoc network using a Raspberry Pi 3B+. The portability and low power consumption of the Raspberry Pi make it ideal for deploying AndroCom on-the-go whenever communication needs arise.

**Table of Contents**

[**ACKNOWLEDGEMENTS** 4](#_Toc165233709)

[Chapter 1 11](#_Toc165233710)

[Introduction 11](#_Toc165233711)

[1.1. Project Introduction 11](#_Toc165233712)

[1.2. Existing Examples / Solutions 11](#_Toc165233713)

[1.3. Business Scope 12](#_Toc165233714)

[1.4. Useful Tools & Technologies 13](#_Toc165233715)

[1.5. Project Work Break Down 15](#_Toc165233716)

[1.6. Project Time Line 16](#_Toc165233717)

[Chapter 2 17](#_Toc165233718)

[Requirement Specification and Analysis 17](#_Toc165233719)

[Requirement Specification **Error! Bookmark not defined.**](#_Toc165233720)

[2.1. Functional Requirements **Error! Bookmark not defined.**](#_Toc165233721)

[2.2. Non-Functional Requirements **Error! Bookmark not defined.**](#_Toc165233722)

[2.3. Selected Functional Requirements **Error! Bookmark not defined.**](#_Toc165233723)

[2.4. System Use Case Modeling **Error! Bookmark not defined.**](#_Toc165233724)

[2.5. System Sequence diagrams 29](#_Toc165233725)

[2.6. Domain Model 29](#_Toc165233726)

[Chapter 3 31](#_Toc165233727)

[System Design 31](#_Toc165233728)

[3.1. Software Architecture 32](#_Toc165233729)

[3.2. Class Diagram 33](#_Toc165233730)

[3.3. Sequence Diagram 33](#_Toc165233731)

[3.4. Entity Relationship Diagram 34](#_Toc165233732)

[3.5. Database Schema 35](#_Toc165233733)

[3.6. User Interface Design 36](#_Toc165233734)

[3.7. Software COTS 38](#_Toc165233735)

[Chapter 4 39](#_Toc165233736)

[Software Development 39](#_Toc165233737)

[4.1. Coding Standards 39](#_Toc165233738)

[4.2. Development Environment 39](#_Toc165233739)

[4.3. Software Description 40](#_Toc165233740)

[Chapter 5 42](#_Toc165233741)

[Software Testing 42](#_Toc165233742)

[5.1. Testing Methodology 42](#_Toc165233743)

[5.2. Testing Environment 42](#_Toc165233744)

[5.3. Test Cases 42](#_Toc165233745)

[6.1. Installation / Deployment Process Description 43](#_Toc165233746)

[Chapter 7 44](#_Toc165233747)

[Project Evaluation 44](#_Toc165233748)

[7.1. Project Evaluation Report 44](#_Toc165233749)

[Book 45](#_Toc165233750)

**List of Figures**

Figure 1.1: Raspberry Pi 3B+ 14

Figure 1.2: Project Work Breakdown 15

Figure 1.3: Project Timeline 16

Figure 2.1: System Use Case Diagram 16

Figure 2.2: System Sequence Diagram 1 (configure Raspberry Pi) 16

Figure 2.3: System Sequence Diagram 2 (network authentication) 16

Figure 2.1: System Use Case Diagram 16

Figure 2.1: System Use Case Diagram 16

**Figure 1.2: Sample Gant Chart**

**Figure 2.1: Sample Use case Diagram**

**Figure 2.2: System Sequence Diagram**

**Figure 2.3: Domain Model**

**Figure 3.1: Software Architecture Diagram**

**Figure 3.2: UML Class Diagram**

**Figure 3.3: Sequence Diagram**

**Figure 3.4: Entity Relationship Diagram**

**Figure 3.5: Database Schema**

**Figure 3.6: Common GUI elements**

**Figure 3.7: Example Login Page UI Design with description in text**

**List of Tables**

Table 1.1: Existing Solutions Feature Comparison 10

Table 2.1: Functional Requirements 17

Table 2.2: Non-Functional Requirements 18

Table 2.3: Selected Functional Requirements 18

Table 2.4: Use Case 1 (Configure Raspberry Pi for ad hoc network) 18

Table 2.5: Use Case 2 (User profile setup) 18

Table 2.6: Use Case 3 (List active users connected with network) 18

Table 2.7: Use Case 4 (Text messaging) 18

Table 2.8: Use Case 5 (Block or unblock users) 18

Table 2.9: Use Case 6 (Voice call) 18

Table 2.10: Use Case 7 (Mute notifications) 18

Table 2.11: Use Case 8 (Video call) 18

Table 2.12: Use Case 9 (Mute mic or turn off camera) 18

**Table 1.2: Raspberry Pi 3B+ Specifications 14**

**Table 1.1: Existing Solutions Feature Comparison 10**

**Table 1.1: Existing Solutions Feature Comparison 10**

**Table 1.1: Existing Solutions Feature Comparison 10**

**Table 1.1: Existing Solutions Feature Comparison 10**

**Table 2.2: Functional and Non-Functional Requirement**

**Table 2.1: Use Case 1**

**Table 2.2: Use Case 1**

**Table 3.1: Data Dictionary**

This page is kept blank

# 

# Chapter 1

# Introduction

## Project Introduction

Our reliance on the internet for communication creates a major vulnerability: outages can cripple our ability to connect. AndroCom, an Android app, offers a solution. AndroCom allows users to send encrypted text messages, make voice calls, and even video calls – all without relying on an internet connection. This unique feature makes it ideal for situations like remote locations with no internet access or as a reliable backup during emergencies.

The secret behind AndroCom's functionality lies in its ability to create an ad hoc network. An ad hoc network is a temporary, decentralized network set up on the fly without needing any existing infrastructure like routers or access points. It utilizes a Raspberry Pi, a small and portable computer, to establish an "ad hoc" network. This network allows devices running the AndroCom app to communicate directly with each other.

AndroCom goes a step further by ensuring the security of your communications. Using AES (Advanced Encryption Standard), it protects your messages, calls, and videos from unauthorized access. This makes it a valuable tool for situations where privacy is paramount. The combination of offline functionality, portability, and secure communication makes AndroCom a versatile and important tool for various situations.

## Existing Examples / Solutions

Even though there isn't a direct equivalent to AndroCom, several technologies do exist that provide the ability to communicate without internet access or existing network infrastructure. First, we have satellite communication systems. With a long history, they offer reliable connections in remote areas. However, their cost, limited bandwidth, and dependence on specialized equipment make them less than ideal for everyone. Additionally, bad weather can disrupt signal quality.

Mesh networks, a more recent development, provide an alternative to traditional Wi-Fi. They distribute the internet connection through interconnected devices, extending coverage. While offering redundancy and wider reach, setting up a mesh network can be more complex. Furthermore, their overall range may still be limited compared to satellite communication.

Finally, walkie-talkies and short-range radios represent some of the oldest forms of off-grid communication. They're simple, reliable for voice communication within a short range, and readily available. However, they lack the ability to transmit data like text messages or video calls, and physical barriers or distance can disrupt communication.

Existing solutions clearly have limitations. AndroCom steps in to bridge these gaps. By leveraging affordable hardware like Raspberry Pi, it offers a more cost-effective approach compared to satellite communication. Setting up an ad hoc network with AndroCom is simpler than configuring mesh networks or requiring specialized satellite equipment. Additionally, AndroCom goes beyond voice communication, enabling text messaging, voice calls, and even video calls, providing a richer communication experience. Finally, the compact size of the Raspberry Pi allows for easy deployment on-the-go, making AndroCom more versatile than fixed solutions like satellite communication. Through these improvements, AndroCom aims to provide a more accessible, user-friendly, and feature-rich solution for communication during internet outages.

**Table 1.1: Existing Solutions Features Comparison**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Feature** | **Satellite Communication System** | **Mesh Network** | **Walkie-Talkie** | **AndroCom Ad Hoc Network (Proposed System)** |
| 1 | Cost Effective |  |  | ✓ | ✓ |
| 2 | Long Range | ✓ | ✓ |  |  |
| 3 | Portable |  |  | ✓ | ✓ |
| 4 | Scalable | ✓ | ✓ |  | ✓ |
| 5 | Security | ✓ |  |  | ✓ |
| 6 | Quality |  |  |  | ✓ |
| 7 | Text Messages | ✓ | ✓ |  | ✓ |
| 8 | Voice Call | ✓ |  | ✓ | ✓ |
| 9 | Video Call |  |  |  | ✓ |

## Business Scope

AndroCom presents a significant business opportunity by offering a unique and cost-effective solution for communication during internet outages. Beyond its initial target audience, AndroCom has the potential to serve various commercial markets. With minimal additional resources, such as user interface refinement and marketing materials, AndroCom can be deployed as a successful commercial product.

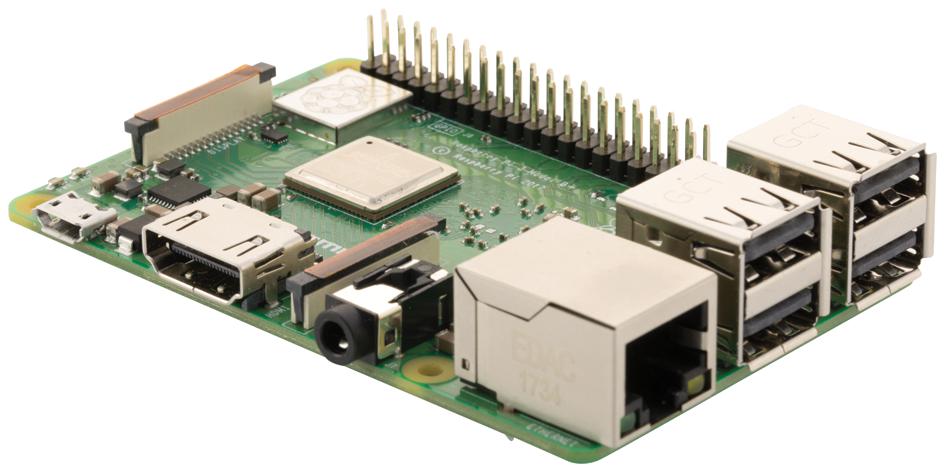
## Useful Tools & Technologies

Following is a list of technologies that are used for designing, development and testing phases of the project.

* + 1. **Programming Languages:**
* **Bash:** Bash is a type of shell scripting language for Unix-based operating systems. It's used to execute commands from the command line and automate tasks.
* **Kotlin:** Kotlin is a general-purpose programming language initially developed by JetBrains for the Android platform. It's known for being concise, interoperable with Java, and supporting functional programming features.
* **Java:** Java is a general-purpose, class-based, object-oriented programming language that is widely used for developing applications of various kinds. It's known for its platform independence and code reusability.
* **Python:** Python is a general-purpose, interpreted, high-level programming language that's known for its readability and ease of use. It's widely used for web development, data science, scripting, and more.
  + 1. **Markup Language:**
* **XML:** XML is a markup language designed for encoding documents in a machine-readable format. It's used for data interchange between applications and for configuring settings in software programs.
  + 1. **Database:**
* **SQLite:** SQLite is a relational database management system (RDBMS) that is embedded directly into the application. It's a lightweight, self-contained, open source database that doesn't require a separate server process.
  + 1. **Development Tools:**
* **Android Studio:** Android Studio is an integrated development environment (IDE) specifically built for Android app development. It provides a comprehensive set of tools for designing, developing, testing, and debugging Android apps.
* **Thonny IDE:** Thonny is a free and open-source Python IDE designed for beginners. It provides a user-friendly interface with features like syntax highlighting, code completion, and debugging tools.
* **IntelliJ IDEA:** IntelliJ IDEA is an IDE from JetBrains that supports various programming languages, including Java and Kotlin. It offers a wide range of features for development, testing, and debugging applications.
  + 1. **Design Tool:**
* **Figma:** Figma is a web-based design tool used for creating user interfaces and user experiences for web and mobile applications.
  + 1. **Hardware:**
* **Raspberry Pi 3B+:** Raspberry Pi is a series of small single-board computers, it's a popular platform for hobbyists and makers due to its affordability and wide range of capabilities.

**Table 1.2: Raspberry Pi 3B+ Specifications**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Specification** | |
| 1 | Processor | 1.4 GHz 64-bit quad-core ARM Cortex-A53 CPU (BCM2837B0) |
| 2 | Memory | 1 GB LPDDR2 SDRAM |
| 3 | Graphics | VideoCore IV 3D graphics core @ 400MHz/300MHz |
| 4 | Storage | MicroSD card slot |
| 5 | Connectivity | Gigabit Ethernet port  Dual-band Wi-Fi (2.4 GHz and 5 GHz) IEEE 802.11.b/g/n/ac  Bluetooth 4.2 Low Energy (BLE)  4 x USB 2.0 ports  40-pin GPIO header for connecting various electronic components |
| 6 | Power | Micro USB power supply (5V, 2.5A) |



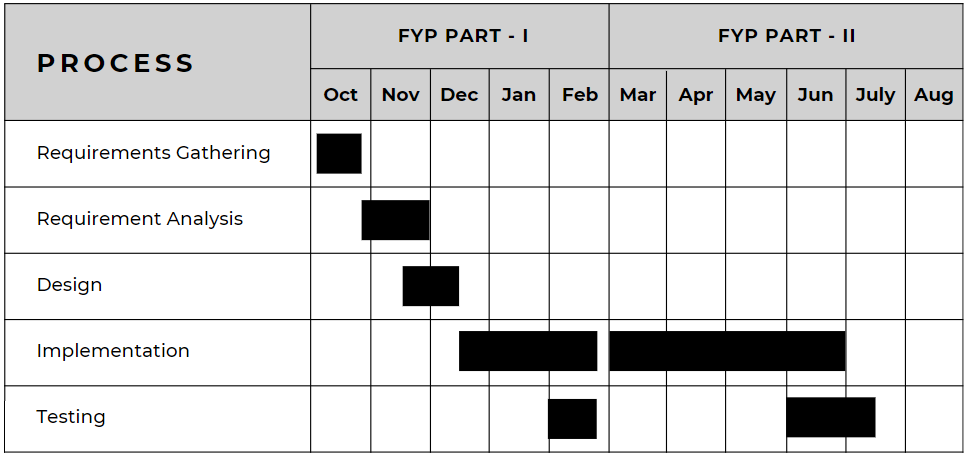
**Figure 1.1: Raspberry Pi 3B+**

## Project Work Break Down

A project work breakdown diagram is a way to break down a complex project into smaller, more manageable tasks. The project work breakdown for the AndroCom is given in Figure 1.2.

**Figure 1.2: Project Work Breakdown**

## Project Time Line

A project timeline diagram is a visual representation of the tasks and milestones in a project, showing their start and end dates. In simple words, it's a bar chart that shows when things need to happen in order for your project to finish on time. The project timeline for AndroCom is given in Figure 1.3.

**Figure 1.3: Project Time Line**

# Chapter 2

# Requirement Specification and Analysis

Requirement analysis in software development is the process of understanding what the software needs to do. This involves gathering information from stakeholders, analyzing that information, and documenting the requirements in a clear and concise way. This Chapter documents the specification and analysis of requirements for AndroCom. Furthermore, it covers the following specifications for the required software:

* Functional & Non-Functional Requirements
* Use Case Diagram
* Brief Description of Each Use Case
* Detailed Sequence Diagram for Each Use Case
* Domain Model
* System Architecture

## Functional Requirements

Functional requirements are what a system must do to meet user needs. The functional requirements for AndroCom are given in Table 2.1.

**Table 2.1: Functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Functional Requirement** | **Type** | **Status** |
| 1 | Configure Raspberry Pi for ad hoc network | Core | Complete |
| 2 | User profile setup in android application | Core | Complete |
| 3 | Broadcast active user list to android application | Core | Complete |
| 4 | Block or unblock users | Intermediate | Complete |
| 5 | Mute notifications | Intermediate | Complete |
| 6 | Text messaging | Core | Complete |
| 7 | Voice Calling | Core | Complete |
| 8 | Video Calling | Core | Pending |
| 9 | Mute mic or disable camera within a call | Intermediate | Pending |

## Non-Functional Requirements

Non-functional requirements are the constraints on how a system should work. The non-functional requirements for AndroCom are given in Table 2.2.

**Table 2.2: Functional and Non-Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Non Functional Requirements** | **Category** |
| 1 | Prompt when connected to the wrong network | Security |
| 2 | End-to-end text encryption | Security |
| 3 | User friendly UI | Usability |
| 4 | Optimize resource usage on Raspberry Pi | Performance |
| 5 | Low power consumption and lightweight android application | Performance |
| 6 | Optimize ad hoc network usage | Performance |

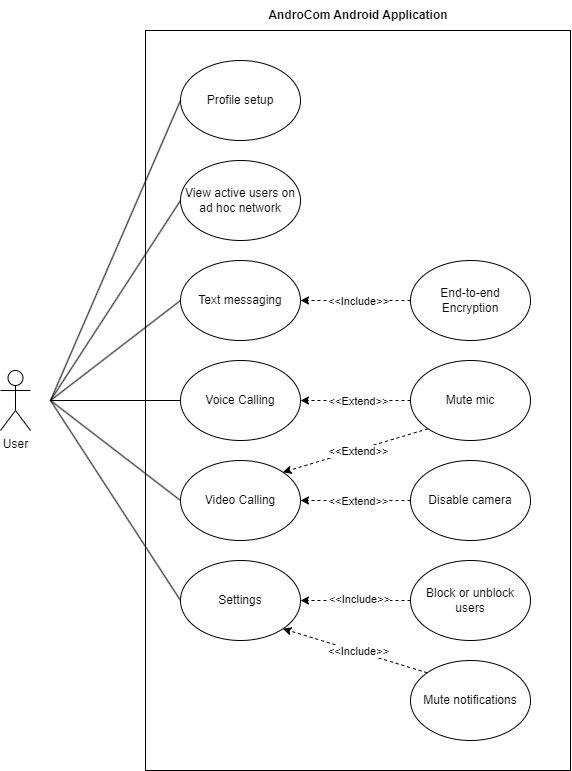
## Selected Functional Requirements

The Selected function requirements of AndroCom for FYP Part-II are given in Table 2.3.

**Table 2.3: Selected Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Functional Requirement** | **Type** |
| 1 | Block or unblock users | Intermediate |
| 2 | Mute notifications | Intermediate |
| 3 | Voice Calling | Core |
| 4 | Video Calling | Core |
| 5 | Mute mic or disable camera within a call | Intermediate |

## System Use Case Modeling

****A system use case diagram is a visual representation of the different ways that users can interact with a system. The system use case diagram of AndroCom is shown in Figure 2.1.

**Figure 2.1: System Use Case Diagram**

**Use Case 1 (Configure Raspberry Pi for Ad Hoc Network):**

**Table 2.4: Use Case 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC1 | | | | |
| **Use Case Name:** | Configured Microcontroller for network connection | | | | |
| **Created By:** | Wasia | | **Last Updated By:** | | Muhammad Harris |
| **Date Created:** | 22-10-2023 | | **Last Revision Date:** | | 29-10-2023 |
| **Actors:** | | User | | | |
| **Description:** | | The user will turn on the microcontroller and connect with its network on his android device | | | |
| **Trigger:** | | The user wants to create an ad hoc network | | | |
| **Preconditions:** | | The user has no internet and wants to use AndroCom for his communication | | | |
| **Post conditions:** | | An ad hoc network will be created via the microcontroller | | | |
| **Normal Flow:** | | User | | System | |
| 1.User turns on the microcontroller | | Microcontroller creates an ad hoc network | |
|  | | 2.User connects with ad hoc network via his android device | | User is connected with the network and is allowed to communicate on the network | |
| **Alternative Flows:** | | None | | | |
| **Exceptions:** | | 1. Microcontroller doesn’t turn on.  2. No network is created by the microcontroller.  3. Network doesn’t show on available networks list. | | | |

**Use Case 2 (User Profile Setup):**

**Table 2.5: Use Case 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC2 | | | | |
| **Use Case Name:** | User profile setup | | | | |
| **Created By:** | Umer Ahmed | | **Last Updated By:** | | Wasia |
| **Date Created:** | 22-10-2023 | | **Last Revision Date:** | | 30-10-2023 |
| **Actors:** | | User | | | |
| **Description:** | | The user will be presented with a sign-up screen where he enters their first name, last name and optionally uploads their profile picture in app. | | | |
| **Trigger:** | | The user installs the application and wishes to complete their profile setup. | | | |
| **Preconditions:** | | The user has successfully installed the application and is on the initial sign-up screen. | | | |
| **Post conditions:** | | The user’s profile information is saved and he can access, use their profile within application. | | | |
| **Normal Flow:** | | User | | System | |
| 1.User clicks get started button to request for sign-up. | | The system provides a User initial page for profile setup. | |
|  | | 2.User provide first name, last name and clicks continue. | | The system re-direct the User to a newly created profile page. | |
| **Alternative Flows:** | | The user cancels the profile setup. | | | |
| **Exceptions:** | | 1. The User has not filled the form correctly.  2. The system is not responding. | | | |

**Use Case 3 (List of active users connected with network):**

**Table 2.6: Use Case 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC3 | | | | |
| **Use Case Name:** | List of active users connected with network | | | | |
| **Created By:** | Wasia | | **Last Updated By:** | | Umer Ahmed |
| **Date Created:** | 22-10-2023 | | **Last Revision Date:** | | 30-10-2023 |
| **Actors:** | | User | | | |
| **Description:** | | System will provide a list of active users who are currently connected to the network for monitoring and management purposes. | | | |
| **Trigger:** | | The user selects the “Active Users” option from the application menu. | | | |
| **Preconditions:** | | User is logged into the application and a connection is established. | | | |
| **Post conditions:** | | The application displays a list of all active users connected to the network. | | | |
| **Normal Flow:** | | User | | System | |
| 1. The user selects the “Active Users” option from the application menu. | | The system displays the list of active users to the user. | |
| **Alternative Flows:** | | Network connection is not established and error message will be displayed. | | | |
| **Exceptions:** | | Network server is unavailable. | | | |

**Use Case 4 (Text messaging):**

**Table 2.7: Use Case 4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC4 | | | | |
| **Use Case Name:** | Text messages with active users | | | | |
| **Created By:** | Muhammad Harris | | **Last Updated By:** | | Wasia |
| **Date Created:** | 22-10-2023 | | **Last Revision Date:** | | 30-10-2023 |
| **Actors:** | | User | | | |
| **Description:** | | User can send and receive text messages to other active users on the network. | | | |
| **Trigger:** | | The user selects the chat icon from the application menu. | | | |
| **Preconditions:** | | The user must be connected to network and is logged into the application. | | | |
| **Post conditions:** | | The user is able to send and receive text messages to other active users on a network. | | | |
| **Normal Flow:** | | User | | System | |
| 1. The user selects the chat icon from application menu. | | The system displays the chat section the application. | |
|  | | 2.The user selects the recipient of a text message. | | The system highlights selected recipient. | |
|  | | 3. The user enters the text message and sends it. | | The system send text message to the recipient over network. | |
| **Alternative Flows:** | | Recipient is not active and text message is not delivered. | | | |
| **Exceptions:** | | 1. User not logged into the application  2. Network server is unavailable. | | | |

**Use Case 5 (Block or unblock users):**

**Table 2.8: Use Case 5**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC5 | | | | |
| **Use Case Name:** | Block and unblock users | | | | |
| **Created By:** | Umer Ahmed | | **Last Updated By:** | | Wasia |
| **Date Created:** | 22-10-2023 | | **Last Revision Date:** | | 30-10-2023 |
| **Actors:** | | User | | | |
| **Description:** | | User can block and unblock other user within application. Blocking user will prevent further communication from or to the blocked user. | | | |
| **Trigger:** | | The user selects the “Block or unblock users” option from the application menu. | | | |
| **Preconditions:** | | User must be connected to network and is logged into the application interacting with the user whose status they want to change. | | | |
| **Post conditions:** | | The selected user is either blocked or unblocked, as per the user’s action. | | | |
| **Normal Flow:** | | User | | System | |
| 1. The user selects the “Block or unblock users” from application menu. | | The system displays a list of all active users on the network. | |
|  | | 2.The user selects the user they want to block or unblock. | | The system blocks or unblocks the selected user. | |
| **Alternative Flows:** | | User is already blocked or selected user is not active. | | | |
| **Exceptions:** | | 1. User not logged into the application.  2.Network connection is not established. | | | |

**Use Case 6 (Voice call):**

**Table 2.9: Use Case 6**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC6 | | | | |
| **Use Case Name:** | Voice call with active users | | | | |
| **Created By:** | Umer Ahmed | | **Last Updated By:** | | Muhammad Harris |
| **Date Created:** | 10-10-2023 | | **Last Revision Date:** | | 11-10-2023 |
| **Actors:** | | User | | | |
| **Description:** | | User can initiate and receive a voice call from an active user within application. | | | |
| **Trigger:** | | The user will press the call icon from the text chat section. | | | |
| **Preconditions:** | | User must be connected to network and is logged into the application. The user has selected an active user from the list and indicated to make a voice call. | | | |
| **Post conditions:** | | The user is able to a voice call with another active user on the network. | | | |
| **Normal Flow:** | | User | | System | |
| 1. The user selects the recipient of the voice call and initiates the voice call. | | The system sends a voice call request to the recipient over the network. | |
|  | | 2. The users are able to talk to each other over the voice connection. | | The system establishes a voice connection between two users. | |
| **Alternative Flows:** | | The recipient user rejects the voice call request and call is not established. | | | |
| **Exceptions:** | | 1. User not logged into the application.  2.Selected active user is no longer available or active. | | | |

**Use Case 7 (Mute notifications):**

**Table 2.10: Use Case 7**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC7 | | | | |
| **Use Case Name:** | Mute notifications | | | | |
| **Created By:** | Wasia | | **Last Updated By:** | | Muhammad Harris |
| **Date Created:** | 10-10-2023 | | **Last Revision Date:** | | 19-10-2023 |
| **Actors:** | | User | | | |
| **Description:** | | User can mute messages and call notifications of specific users within application. | | | |
| **Trigger:** | | The user selects the “Mute messages and call notification” option from the settings or user preferences section. | | | |
| **Preconditions:** | | User must be connected to network and is logged into the application. Also, user has identified specific users from whom they want to mute notifications. | | | |
| **Post conditions:** | | The selected user’s message and call notifications are either muted or unmuted, as per user’s action. | | | |
| **Normal Flow:** | | User | | System | |
| 1. The user selects the “Mute messages and call notifications” option. | | The system displays a list of all active users on the network. | |
|  | | 2. The user selects the users whose messages and call notifications they want to mute. | | The system mutes messages and call notifications from the selected user. | |
| **Alternative Flows:** | | The selected user is already muted or a user is not active. | | | |
| **Exceptions:** | | 1. User not logged into the application and is not connected to the network. | | | |

**Use Case 8 (Video call):**

**Table 2.11: Use Case 8**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC8 | | | | |
| **Use Case Name:** | Video call with active users | | | | |
| **Created By:** | Wasia | | **Last Updated By:** | | Muhammad Harris |
| **Date Created:** | 21-4-2024 | | **Last Revision Date:** | | 25-4-2024 |
| **Actors:** | | User | | | |
| **Description:** | | User can initiate and receive a video call from an active user within application. | | | |
| **Trigger:** | | The user will press the video call icon from the text chat section. | | | |
| **Preconditions:** | | User must be connected to network and is logged into the application. The user has selected an active user from the list and indicated to make a video call. | | | |
| **Post conditions:** | | The user is able to make a video call with another active user on the network. | | | |
| **Normal Flow:** | | User | | System | |
| 1. The user selects the recipient of the video call and initiates the video call. | | The system sends a video call request to the recipient over the network. | |
|  | | 2. The users are able to talk to each other over the video connection. | | The system establishes a video connection between two users. | |
| **Alternative Flows:** | | The recipient user rejects the video call request and call is not established. | | | |
| **Exceptions:** | | 1. User is not logged into the application.  2. Select active user. | | | |

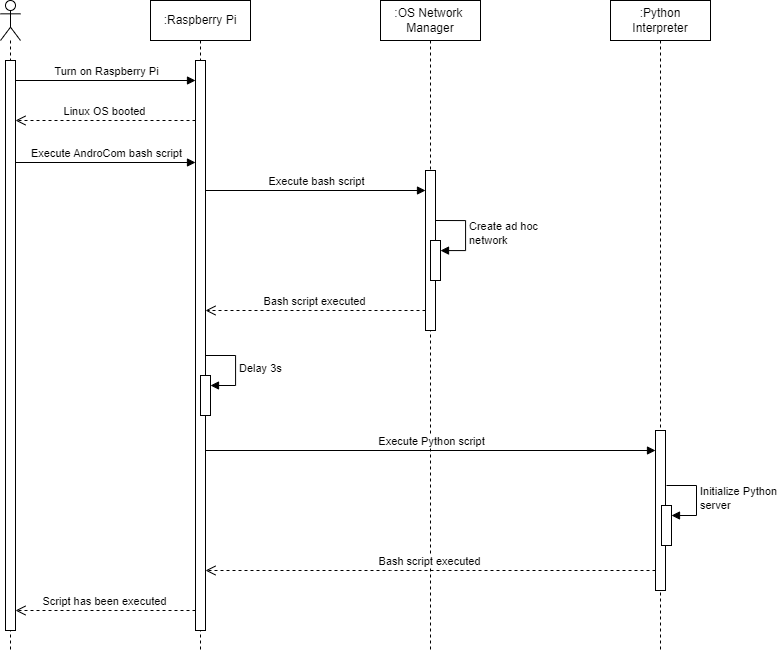
**Use Case 9 (Mute mic or turn-off camera):**

**Table 2.12: Use Case 9**

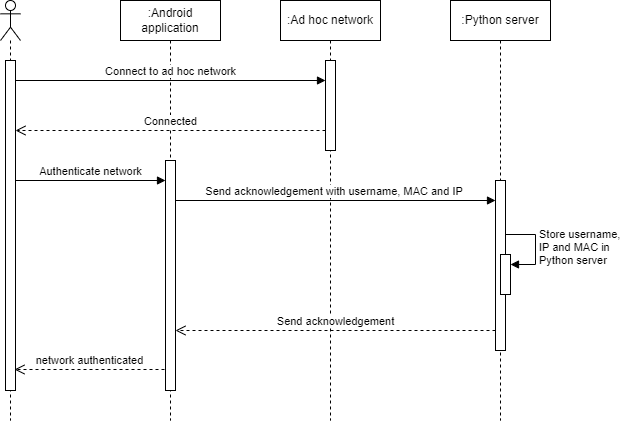
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC9 | | | | |
| **Use Case Name:** | Mute mic and turn-off the camera | | | | |
| **Created By:** | Umer Ahmed | | **Last Updated By:** | | Muhammad Harris |
| **Date Created:** | 21-4-2024 | | **Last Revision Date:** | | 25-4-2024 |
| **Actors:** | | User | | | |
| **Description:** | | User mutes their microphone and turn-off the camera to avoid unwanted noise and to maintain focus on call content. | | | |
| **Trigger:** | | The user will press the mute mic and turn-off the camera icon on the call screen. | | | |
| **Preconditions:** | | The user must be connected to the network and is logged into the application. The user is participating in a call. | | | |
| **Post conditions:** | | The user’s microphone and camera can be independently muted or turned off, and no audio or video is transmitted by the user’s device. | | | |
| **Normal Flow:** | | User | | System | |
| 1. The user decides to mute the microphone or turn-off the camera. | | The system provides both microphone mute and turn-off camera buttons to the user. | |
|  | | 2. The user either clicks the mute or turn-off camera button. | | The system confirms the actions of user. | |
| **Alternative Flows:** | | The user clicks the mute or turn-off camera button but the microphone remains unmuted and the camera isn’t turned off. | | | |
| **Exceptions:** | | The call with another user crashes and the mute mic or turn off camera functionality becomes unavailable. | | | |

## System Sequence Diagrams

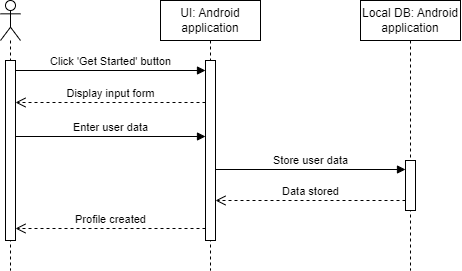
Sequence diagrams are created to show the sequence of events among user and the system to complete an action / use case. Following are the system sequence diagrams for AndroCom.



**Figure 2.2: System Sequence Diagram 1 (Configure Raspberry Pi)**

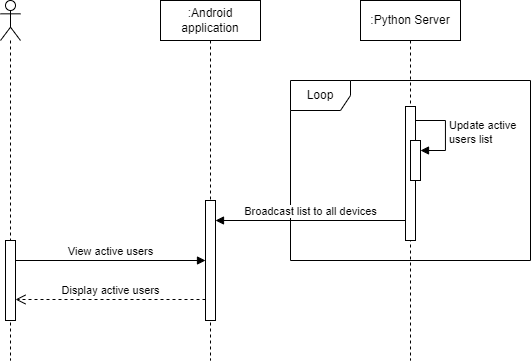


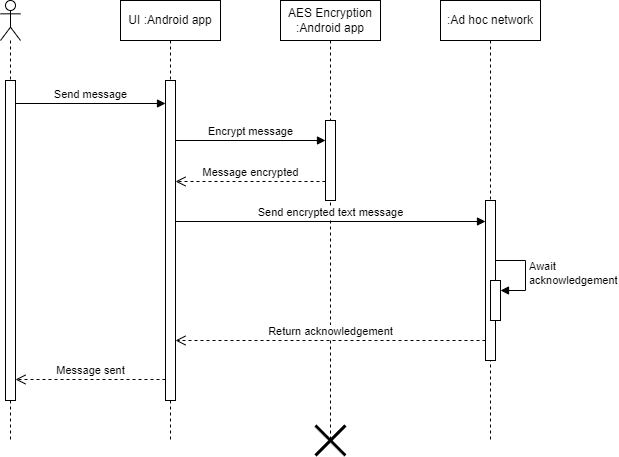
**Figure 2.3: System Sequence Diagram 2 (Network authentication)**



**Figure 2.4: System Sequence Diagram 3 (Profile setup)**

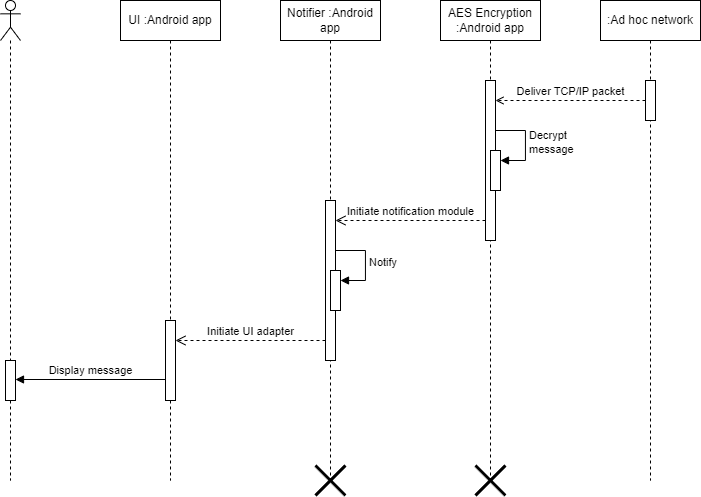
**Figure 2.5: System Sequence Diagram 4 (Broadcast active user list)**

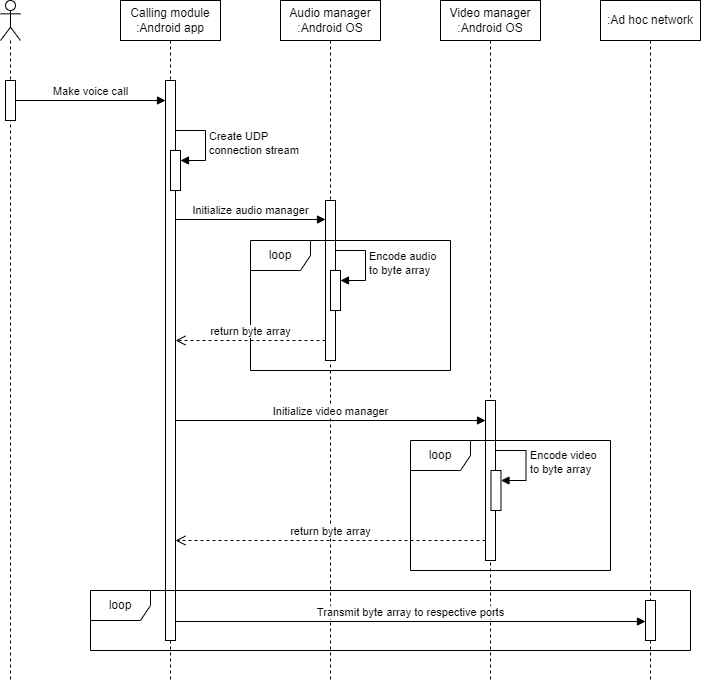




**Figure 2.6: System Sequence Diagram 5 (Send text message)**

**Figure 2.7: System Sequence Diagram 6 (Receive text message)**

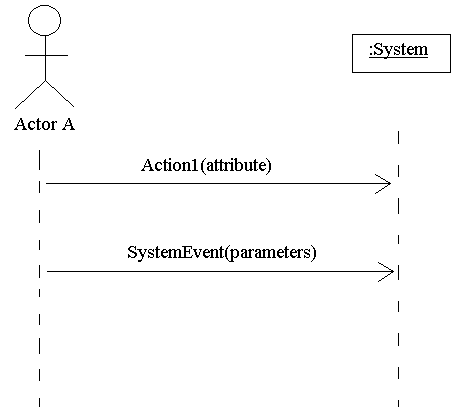


**Figure 2.8: System Sequence Diagram 7 (Make voice or video call)**

## System Sequence diagrams

Sequence diagrams are created to show the sequence of events among user and the system to complete an action / use case. A sample is presented in Fig 2.2.

*You are required to provide SSD of all the uses cases that you have provided above.*

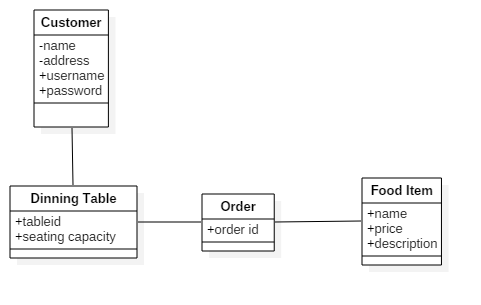


**Figure 2.2: System Sequence Diagram**

## Domain Model

Part of your initial architectural modeling efforts, particularly for a business application, will likely include the development of high-level domain model as you see in Fig. 2.3. This model should be very slim, capturing the main business entities and the relationships between them. Some people consider this type of model to be

an initial requirements model instead of an initial architecture model



**Figure 2.3: Domain Model**

# Chapter 3

# System Design

The purpose of this chapter is to provide information that is complementary to the code. Without an adequate design that delivers required function as well as quality attributes, the project will fail. But communicating architecture to its stakeholders is as important a job as creating it in the first place.

There are two views that are considered while defining software architecture. There are specific design artifacts that belong to each view. Description of such artifacts is given below. *You may select the artifacts depending on the nature of your project.*

* Structural View
  + Architecture diagram
  + Module structure diagram
  + Component diagram
  + Class diagram
* Behavioral View
  + Sequence diagram
  + Activity diagram
  + State machine diagram

At a high level, a software architecture document includes:

1. An outline description of the software design, including major software components and their interactions.
2. A common understanding of requirements, constraints and principles that influence the architecture.
3. A description of the hardware and software platforms on which the system is built and deployed.
4. Explicit justification of how the architecture satisfies the above mentioned points.

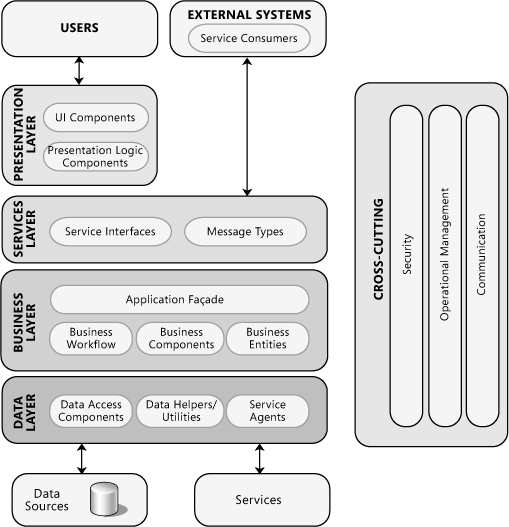
Design pattern is a description or template for how to solve a problem that can be used in many different situations. Object-oriented design patterns typically show relationships and interactions between classes or objects, without specifying the final application classes or objects that are involved.

*It is important that you justify its design, for example, by discussing the implications of constraints on your solution and different design choices, and then giving reasons for making the choices you did. At each stage of the design you should mention what kind of design patters have you followed while designing your system. You should identify which design pattern among the existing patterns are you following while designing your project.*

## Software Architecture

Software architecture is described as the organization or structure of a system, where the system represents a collection of components that accomplish a specific function or set of functions. When getting started with your design, keep in mind the key principles that will help you to create an architecture that adheres to proven principles, minimizes costs and maintenance requirements, and promotes usability and extendibility. The major decisions that you must make, and which help to ensure that you consider all of the important factors as you begin and then iteratively develop your architecture design are

* Determine the Application Type
* Determine the Deployment Strategy
* Determine the Appropriate Technologies
* Determine the Quality Attributes
* Determine the Crosscutting Concerns

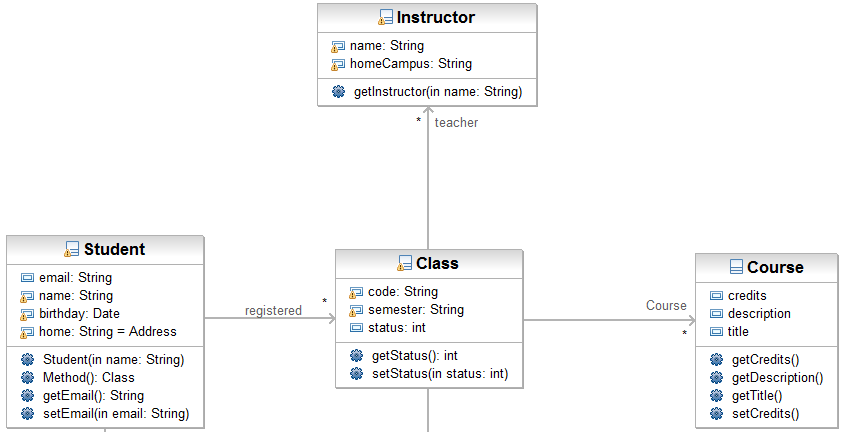


**Figure 3.1: Software Architecture Diagram**

*Fig. 3.1 illustrates common application architecture with components grouped by different areas of concern. You should think of architecture as the strategy for how you will build the system. An architectural “layer” is the top-level logical view, or an abstraction, of your design.*

## Class Diagram

Class Diagram as shown in Fig. 3.2 provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages; from modeling the domain-specific data structure to detailed design of the target system.

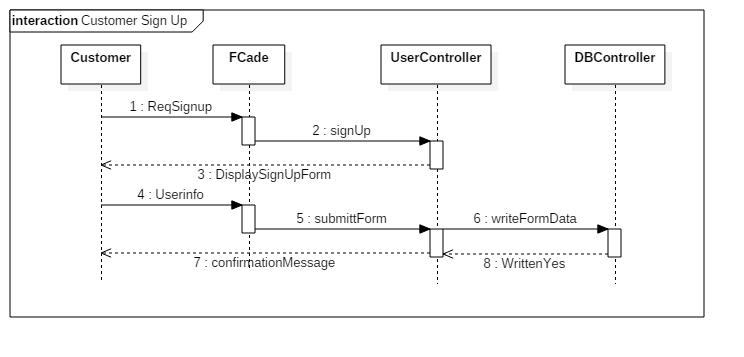


**Figure 3.2: Class Diagram**

## Sequence Diagram

Sequence diagrams, when used in conjunction with class diagrams; provide an extremely effective communication mechanism. UML sequence diagrams as shown in Fig. 3.3 are used to show how objects interact in a given situation.

*You can use a class diagram to illustrate the relationships between the classes, and the sequence diagram lets you show the messages sent among the instances of these classes and the order in which they are sent. When an object sends a message to another object, it implies that the two classes have a relationship that must be shown on a class diagram.*

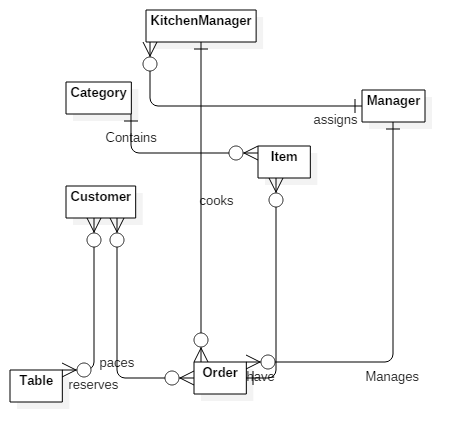


**Figure 3.3: Sequence Diagram**

## 3.4. Entity Relationship Diagram

Entity relationship model diagram (ERD) is a conceptual representation of the data in a software system. During detail design this model is mapped in to the physical database model. There are different diagramming conventions available for creating ER diagrams. A sample ERD is shown in Fig. 3.4.

**.***At this stage you may not be able to provide a comprehensive ERD therefore you may provide the refined ERD later in design chapter of the document where you will provide the fully attributed ERD such as primary keys, cardinality constraints etc.*



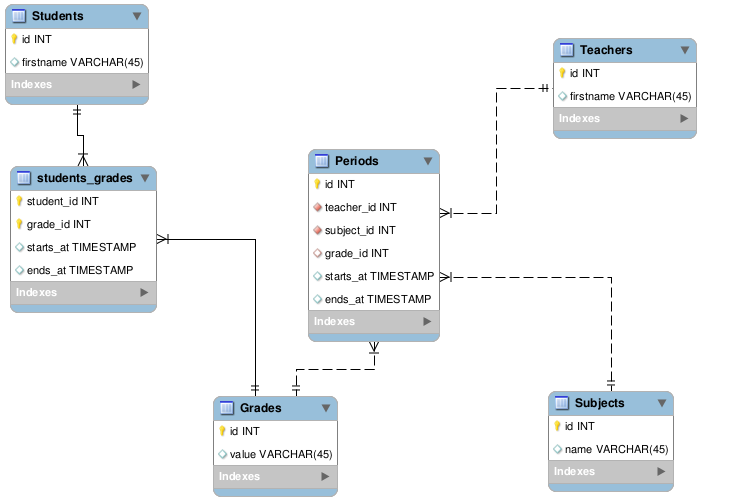
## 

**Figure 3.4: Entity Relationship Diagram**

## Database Schema

A database schema represents the logical configuration of all or part of a relational database. It can exist both as a visual representation and as a set of rules known as integrity constraints that govern a database. These rules are expressed in a data definition language, such as SQL. A database schema indicates how the entities that make up the database relate to one another, including tables, views, stored procedures, and more. A database scheme includes information related to primary and secondary keys, normalization and indexing.

You may present database scheme using front end tool of any DBMS or any other design tools such as Visio or Enterprise Architecture. A sample database scheme is shown in Fig. 3.5.

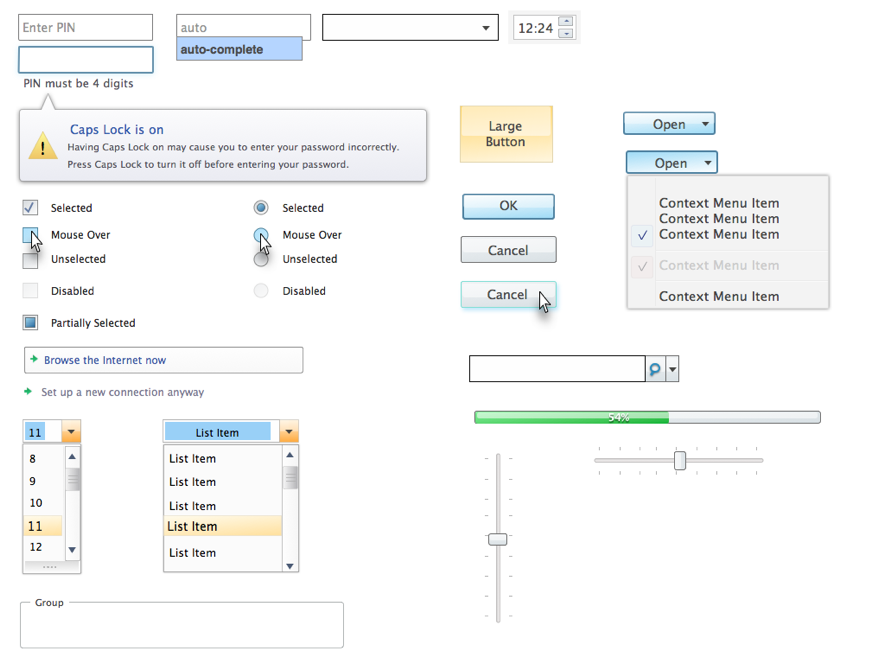
.

**Figure 3.5: Database Schema**

## User Interface Design

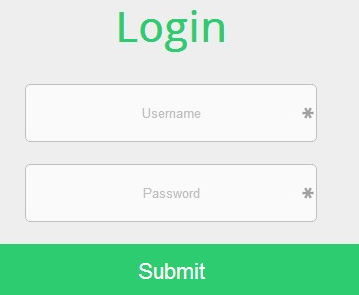
User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. UI brings together concepts from interaction design, visual design, and information architecture.

*You should describe the UI design in such a way that it remains simple and consistent along different views. Common GUI elements are shown in the Fig. 3.6. You should describe the UI design of each page.*



**Figure 3.6: Common GUI elements**

*Example Login Page as shown in Fig. 3.7 will contain one text field and one password field. Max length of text field is 8 and min is 4 whereas maximum length off password is 6 and minimum is 3.*



**Figure 3.7: Example Login Page UI Design with description in text**

## Software COTS

If you use COTS (Components Off the Shelf, also named SOUP, Software Of Unknown Provenance), list them here.

For each COTS, describe:

* Its identification and version
* Its purpose
* Where it comes from: manufacturer, vendor, university …
* Whether it is maintained by a third party or not
* If this is an executable,
  + What are the hardware / software resources it uses
  + Whether it is insulated in the architecture and why

# Chapter 4

# Software Development

The Implementation section is similar to the Specification and Design section in that it describes the system, but it does so at a finer level of detail, down to the code level. This section is about the realization of the concepts and ideas developed earlier. It can also describe any problems that may have arisen during implementation and how you dealt with them.

**Make sure that the system design corresponds to the implementation of the project. If there is no relationship between design and implementation, it may downgrade your score in FYP.**

You should also mention any unforeseen problems you encountered when implementing the system and how and to what extent you overcame them. Common problems are:

* Difficulties involving existing software, because of e.g.,
  + Its complexity,
  + Lack of documentation;
* Lack of suitable supporting software
* Overambitious project aims.

A seemingly disproportionate amount of project time can be taken up in dealing with such problems. The Implementation section gives you the opportunity to show where that most of the effort has been spent.

## Coding Standards

*Describe the indention, declaration,naming convention and statement standard used while coding the project.*

## Development Environment

In this section you will provide the reason behind using all the existing tools and technologies that you may have used during the development of your project. This includes development environment that you have used. How have you deployed the development environment? What different kind of packages you have used? Are there any third party libraries involved etc?

## Software Description

In this section you will identify major modules of the software that you have produced. You will show the class diagram of these major modules for this section. Typical subheadings of this section can be

**Login process**

**Loading Data**

**Data processing**

**Report generation**

**…..**

Moreover you will also discuss the logic that you have implemented in the code of those modules with the help of code snippets as shown below in the examples. Do not attempt to describe all the code in the system, and do not include large pieces of code in this section.

* Are especially critical to the operation of the system.
* You feel might be of particular interest to the reader for some reason
* Illustrate a nonstandard or innovative way of implementing an algorithm, data structure, etc.

**Snippet 1**

#define SWAP(type, x, y) \

do { \

type temp;

temp = x;

x = y;

y = temp; \

} while(1)

**Description**: This function takes 2 arguments. Then we have an infinite loop that swaps the value of the two passed variables.

You are not allowed to include the complete source code of the software how ever you can include important functions of your major modules to discuss the logic of your code.

**Snippet 2**

#define pop(type, Top) \

type temp;

temp = Top.item;

Top = Top.next;

return temp;

**Description**: This function pops the top of the stack. It places the top pointer to the next item of the stack and return the popped item..

You are not allowed to include the complete source code of the software how ever you can include important functions of your major modules to discuss the logic of your code.

# Chapter 5

# Software Testing

Software Testing is the most crucial part of Software Development Process. It is the investigation or evaluation of a software component, improving them, and finding bugs and defects. Testing is usually done by executing a system in such a way that it identifies any gaps, errors, or missing requirements in contrary to the actual requirements.

## 5.1. Testing Methodology

It is essential to have a testing plan in place to ensure that the product delivered is robust and stable, and is delivered on a predictable timeline.

*In this section you will discuss the reason of various testing techniques that you have used to test the software you have created such as integration testing, component testing and system testing etc.*

## 5.2. Testing Environment

*Describe and discuss the reason to use the selected testing environment.*

## 5.3. Test Cases

*You should describe how you demonstrated that the system works as intended (or not, as the case may be). Include comprehensible summaries of the results of all critical tests that were carried out. You might not have had the time to carry out any full rigorous tests you may not even got as far as producing a testable system. However, you should try to indicate how confident you are about whatever you have produced, and also suggest what tests would be required to gain further confidence*

* **Test Case 1**
* Test case description
* How test case was generated
* Expected result of the test case
* Actual result of the test case

**Table 5.1: Test Case 1**

|  |  |
| --- | --- |
| Date: 06 June 2017 |  |
| *System:* Menu Drive |  |
| *Objective:* View location of delivery boy | *Test ID:*1 |
| *Version:*1 | *Test Type:* Unit testing |
| *Input:*  Longitude=33.7294  Latitude=73.0931 | |
| *Expected Result:*return Islamabad location. | |
| *Actual Result:* passed | |

### 

Chapter 6

**Software Deployment**

## 6.1. Installation / Deployment Process Description

*In this section you have to provide step-by-step guide for the installation of the software produced with the help of screen shots.*

# Chapter 7

# Project Evaluation

This chapter includes the examiners evaluation report, including the points to be revised/included along with the selected requirements in the next iteration.

## 7.1. Project Evaluation Report

* References

All the documents, papers, articles and WebPages that you have taken help from must be cited in the references section

## Book

Author(s), Book *Title*. Place of publication: Publisher, year, volume, page number(s).

Example: [1] W.K. Chen, *Linear Networks and Systems*. Belmont, CA: Wadsworth, 1993, pp. 123-35.

**Webpage**

Author(s) and/or organization, date of publish or date the page was last updated, title of web page document, website address that provides a direct link to the document, and the date you last accessed the document

Example: Winston, J 1999, *A look at referencing,* AAA Educational Services, accessed 20 October 2015, <http://www.aaa.edu.au/aaa.html>. United Nations Web Services 2006,

**Research Paper**

*Author(s), "Article title,* Journal Title*, vol., no., page number(s), Month year.*

Example: [2] G. Pevere, "Infrared Nation, *International Journal of Infrared Design*, vol. 33, pp. 56-99, Jan. 1979.

*If you need to reference any item that is not the the list, you should consult IEEE citation format available at the following link* [*http://library.queensu.ca/book/export/html/5846*](http://library.queensu.ca/book/export/html/5846)

**Appendix**

**User Manual of the software**