**Final Year Project Template**

***Dear Students***

*This guide is prepared to help the students in preparing their final year project reports. Report writing is one of the primary professional responsibility of a practising software engineer. The final report of any project is not just a formality. It is a primary product of the engineering efforts and is often the basis for evaluation of the software engineer’s professional abilities. The report is also a service to the software community who needs the information regarding that particular software. The report should stand on its own and it should include all the necessary sections, targeting at a reader who does not necessarily have any prior knowledge about the project or the technology involved in it.*

***How to follow this guideline?***

*The sections in italics of this format report are instructions that you are supposed to follow. The other sections are meant as information that will help you to meet the minimum requirement of the report. The first portion of this report elaborates on the generic issues involved in report writing such as intellectual property, plagiarism, formatting etc.*

***Who is the Target Reader?***

*The target audience of these project reports is bachelor level CS(SE) student who only has basic knowledge of the computer science field. You should therefore assume that the reader is someone who does not know anything about your project but who may refer your work in his or her own project. Your report should therefore include all the details about the ‘what?’, ‘why?’ and ‘how?’.*

*To ensure the reports are easy to read with consistent format, it is very important that students follow the instructions while writing the report. You have taken several courses to improve your English writing skills during the BS(CS)/BS(SE) program. You are supposed to follow the elements and techniques you have learnt in those courses especially the knowledge you have gained in “Technical Report Writing” should be evident in all versions of your repors.*

***Submission Process***

*It is the reasonability of the project supervisor to make sure that the template is followed during the documentation of the different phases of the project. This is a generic document and it can be changed as per the project requirement but only with the approval of the project supervisor. You should focus on completing the artifacts provided in this guideline without worrying about the size of the report. This document must be submitted to the project coordinator on the due date with proper approval from the supervisor and co-supervisor. Failure to follow this guideline will result in the cancelation of the project. The document will be reviewed by supervisor as well as your English instructor and accepted only when its soundness satisfies both of them. Following portions of the report must be completed at each step of the projects.*

|  |  |
| --- | --- |
| ***Artefact*** | ***Completion Stage*** |
| *Chap 1 of v 1.0* | *Proposal Defense* |
| *Chap 2 , 3 of v1.0* | *Part 1 Mid Term* |
| *v 1.0* | *Part 1 Final* |
| *v 2.0* | *Part II Mid term* |
| *v 3.0* | *Part II Final* |

**Project Types**

There are two types of Final Year Project: the Development Project and the Research project.

* **Development Projects**

The objective of this type of the project is to develop a system that meets a set of user needs. This may take many forms such as a server, a program, a library, a collection of programs, an embedded system, plug-ins, modification to existing software etc.

The focus of this project is on the sound software engineering principles and functionality of the software you have produced. Your project will be evaluated in terms of how well it meets the user needs, how well is it tested and the user friendliness of the interface.

* **Research Project**

The objective of the research-oriented project is to solve a research problem. This may take the form of evaluating the effectiveness of existing solutions, modify existing solutions or developing a new solution to the problem.

The focus of this project is on the sound experimental technique, and evaluating the solution thoroughly. You will do a background research into the domain are and develop a basis of your work. The success of your solution will be evaluated on the basis of understanding and use of experimental methods as well as evaluation methodology of your solution.

*You should discuss this question with your supervisor at the proposal defence stage of the project to decide whether your project is a research oriented or a development project.*

**What is intellectual property?**

Intellectual Property is the term used to describe the outputs of creative endeavor in literary, artistic, industrial, scientific and engineering fields that can be protected under legislation. During the course of your project you may generate some novel work therefore it is necessary to understand the concept of intellectual Property.

**Plagiarism – What is it?**

1. Unauthorized act of copying/reproducing or attempt to copy an idea, writing or invention of another person
2. Extraction of academic data which are the results of research undertaken by another person, such as findings of research, data obtained, whether published or unpublished, without giving due acknowledgement to the original source.
3. Unauthorized translation of the writing of another person from one language to another whether wholly or partly.

***What must be done to avoid plagiarism?***

1. ***Citation and references.***

*Following are the main items that require citation.*

* ***Direct quotes:*** *phrases, sentences, or sections copied directly from a text; cite with quotation marks (use a limited amount of text, not a full text)*

***Example****:* “*Failure to reference appropriately will be considered unethical academic behavior and could result in allegations of misconduct*.” [1]

*The [1] symbol at the end of the quote refers to the first entry in a list of references, as shown at the end of this template under a heading ‘references’.*

* ***Paraphrased text:*** *sections of your writing that are based on research (not common knowledge) but written in your own words (not in quotes*
* ***Facts and Figures:*** *numbers, percentages, and facts that have been collected by an exclusive source (such as during an experiment or poll)*
* ***Theories, methods, and ideas****: any original idea or thought that you find during your research and present in your writing*
* ***Images, graphs, illustrations:*** *always follow copyright rules when using images, including those you find online*

*Paraphrased text, facts and figures, theories, methods and ideas, Images, graphs and illustrationsall should be referenced with symbols [ ] as shown in above example.*

*In order to have an accurate record of what you have researched and therefore an accurate reference, it is important that you write down the details of your sources as you study. You should keep a complete list of references as presented in the last section of this report template.*

1. ***Be familiar in the area that you are talking about.*** *By understanding the subject, you are more likely to write in your own words, rather than restate someone else's definition of this subject. Look for information on the topic you want to write about. This can be on the Internet or in books, although books are almost always more authoritative than the Internet.*
2. ***When in doubt, give credit.*** *Mention the source inside your paraphrase: "According to Richard Feynman, quantum electrodynamics can be described using path integral formulations."*

***Spelling***

*There is no excuse for spelling mistakes in any report as spelling errors create a bad impression. Always use a spell checker; they are invaluable for picking up typographical errors as well as genuine spelling mistakes. Note that spelling checkers cannot detect cases where the wrong word happens to be a real word e.g. from – form. So a careful proof read is necessary.*

***Writing in the third person***

*We would strongly advise to write FYP report in the third person. This provides a greater sense of objectivity and distance as the focus is on what is being said rather than who is saying it. To write in the third person, you write as if you are an outsider reporting on the aims, methods and outcomes of your project, rather than writing as though they are happening to you. First person pronouns such as "I" and "We" are replaced with third person pronouns such as "the project leader/team", "he/she, "it" and "they".*

***General Formatting Guidelines***

*Here are some general formatting guidelines that apply to the entire report:*

* *Use 1- or 1-1/2-inch margins for all four margins of the report. You might want to use a 1-1/2-inch margin at the top and 1-inch margins for the left, right, and bottom.*
* *Use a 1-1/2-inch left margin if your binding uses a lot of space*
* *Generally use double-spaced typing except in those areas where single spacing is shown (for example, in the transmittal letter, descriptive abstract, figure titles, short vertical lists, and items in the information-sources list).*
* *Use one side of the paper only.*

***Headings: Specific Format and Style***

***First-Level Headings***

*Follow these guidelines for first-level headings:*

* *Capitalize each word of first level heading except preposition and article but if they appear as the first word of heading then capitalize them as well.*
* *Use Roman OR Arabic numerals with first-levels.*
* *Either underline the words but not the Roman/Arabic numeral, OR bold the entire heading including the numeral.*
* *Make first-levels centered on the page.*
* *Start a new page whenever you have a first-level heading.*
* *Begin first-levels on the standard first text line of a page.*
* *Leave 3 blank lines between first-levels and the first line of text.*
* *Use 18-font size.*
* *Use decimal numbering system for headings.*

***Second-Level Headings***

*Follow these guidelines for second-level headings:*

* *Capitalize each word in second-levels heading.*
* *Use 16-font size and bold.*
* *Make second-levels flush left.*
* *Leave 2 blank lines between previous text and second-levels.*
* *Leave 1 blank line between second-levels and the following text.*

***Third-Level Headings***

*Follow these guidelines for third-level headings:*

* *Make third-levels sentence-style.*
* *Use bold for third-levels.*
* *Do not make third-levels a grammatical part of sentences that follow.*
* *Use the standard spacing between paragraphs for paragraphs that contain third-levels.*

***Page-Numbering Style***

* *All pages within the front and back covers are, but the page number is not always displayed.*
* *All pages coming before page 1 of the introduction use lowercase Roman numerals.*
* *All pages beginning with page 1 of the introduction use with Arabic numerals.*
* *Page numbers are not displayed on the transmittal letter, title page, and first page of the table of contents, page 1 of the introduction, and the appendix divider page.*
* *There are several choices of pagination style for the main-text pages:*
  + *Center page numbers at the bottom (halfway between the last text line and the bottom edge of the paper).*
  + *Place page numbers in the top right corner (on the right margin, halfway between the top text line and the top edge of the paper). Do not display page numbers on any page with a centered (first-level) heading (display it centered at the bottom).*
* *Some word-processing software causes problems in implementing these pagination guidelines; let your instructor know.*

***Figures and Table Labels***

* *Every figure and table must be labeled and referenced in the text. Label of the figure is placed at the bottom of the figure and list of figures must also be generated. A figure/table without their description cannot convey the intended meaning to the reader. Label of the table is placed at the top of the tables and list of tables must also be generated.*

#### AndroCom: P2P Communication Without Internet

#### Wasia BCS203233

#### Muhammad Harris BCS203193

#### Umer Ahmed BCS203182



**Fall - 2023**

**Supervised By**

**Mr. Bilal Ahmed**

**Department of Computer Science**

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

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Version** | | V X.0 | |  | **NUMBER OF MEMBERS** |  |
|  | | | | | | |
| **TITLE** |  | | | | | |
|  | | | | | | |
| **SUPERVISOR NAME** | | |  | | | |

|  |  |  |
| --- | --- | --- |
| **MEMBER NAME** | **REG. NO.** | **EMAIL ADDRESS** |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **MEMBERS’ SIGNATURES** |  | |
|  |  | **Supervisor’s Signature** |
|  |  |
|  |  |

*Note 1: This paper must be signed by your supervisor*

*Note 2: The soft-copies of your project report, source codes, schematics, and executable should be delivered in a CD*

**APPROVAL CERTIFICATE**

This project, entitled as “Menu Drive (Insert Your Project Title Here) ” has been approved for the award of

**Bachelors of Science in Computer Science**

**Committee Signatures:**

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

(Mr. First Name Surname)

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

(Mr. First Name Surname)

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 thatthis undergraduate project, neither as a whole nor as a part thereof has been copied out from any sources, wherever references have been provided.*

|  |
| --- |
| **MEMBERS’ SIGNATURES** |
|  |
|  |
|  |

**ACKNOWLEDGEMENTS**

*It is usual to thank those individuals who have provided particularly useful assistance, technical or otherwise, during your project. Your supervisor will obviously be pleased to be acknowledged as he or she will have invested quite a lot of time overseeing your progress.*

**DEDICATION**

**This is an optional section**

*In this section you dedicate your project to anybody that you feel motivates you for hard work and putting effort for successful life.*

**Executive Summary**

This should be not more than one page in length (200 words approx.). The summary should allow the reader who is unfamiliar with the work to gain a swift and accurate impression of what the project is about, how it arose and what has been achieved.

*It is recommended, you write this section when the report is finished.*

**Table of Contents**

*This should give a complete list of what the report contains starting with the abstract (the title page is not included in the contents list).*

**Chapter 1**

**Introduction**

1.1. Project Introduction

1.2. Existing Examples / Solutions

1.3. Business Scope

1.4. Useful Tools and Technologies

1.5. Project Work Break Down

1.6. Project Time Line

**Chapter 2**

**Requirement Specification and Analysis**

2.1. [Functional Requirements](#_Toc489349696)

2.2. [Non-Functional Requirements](#_Toc489349697)

2.3. [Selected Functional Requirements](#_Toc489349698)

2.4. [System Use Case Modeling](#_Toc489349699)

2.5. [System Sequence diagram](#_Toc489349700)

2.6. [Domain Model](#_Toc489349703)

**Chapter 3**

**System Design**

3.1.[Layer Definition](#_Toc489349706)

3.1.1. [Presentation Layer](#_Toc489349707)

3.1.2. [Business Logic Layer](#_Toc489349708)

3.1.3. [Database Layer](#_Toc489349708)

[3.2. Software Architecture](#_Toc489349710)

[3.3. Class Diagram](#_Toc489349711)

[3.4. Sequence Diagram](#_Toc489349712)

[3.5. Entity Relationship Diagram](#_Toc489349712)

[3.6. Database Schema](#_Toc489349713)

[3.7. User Interface Design](#_Toc489349714)

**Chapter 4**

**Software Development**

4.1[. Coding Standards](#_Toc489349717)

4.1.1. [Indentation](#_Toc489349706)

4.1.2. [Declaration:](#_Toc489349707)

4.1.3. [Statement Standards](#_Toc489349708)

4.1.4. [Naming Conventions:](#_Toc489349707)

4.2. [Development Environment](#_Toc489349708)

4.3. [Database Management System](#_Toc489349706)

[4.4. Software Description](#_Toc489349718)

**Chapter 5**

**Software Testing**

5.1. [Testing Methodology](#_Toc489349721)

[5.2. Testing Environment](#_Toc489349722)  5.2.1. Test Cases

[5.2. Testing Results](#_Toc489349722)

**Chapter 6**

**Software Deployment**

6.1. Installation/Deploment Process Description

**Chapter 7**

**Project Evaluation**

7.1. Project Evaluation Report

**List of Figures**

*If the report contains figures or tables a list of these should be provided. The list should give the table or figure number, the title of the table or figure and the page number. If only a few tables and figures are present, they may be treated on one page. Remember that all figures and tables used must be referred in the text. For example “The class diagram shown in Figure 2.1 ....”*

**Figure1. 1: Work breakdown Structure**

**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 2.1: Functional Requirements**

**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

In today's world, the internet has become such an integral part of our lives that if it were to go down tomorrow, most forms of communication, such as messaging, calls, and video communication, would cease to function. *AndroCom* is an Android app that enables its users to send text messages, make voice calls, and engage in video calls with complete end-to-end encryption when communicating with other users without the need of an internet connection.

*AndroCom* has significant market potential due to its unique features. It can work in places with no internet, help universities with daily tasks, and serve as a reliable backup during internet outages or emergencies. Its versatility and special capabilities make it valuable in various situations, making it an important tool in the market.

This functionality is implemented through an AD HOC network that is created using a microcontroller, specifically a Raspberry Pi, which serves as a critical component of the system. A server is created on the Raspberry Pi using Python, enabling packet transfer between the Raspberry Pi and the devices using AndroCom. This innovative setup ensures secure and efficient communication while bypassing the need for a traditional internet connection, addressing the challenges posed by internet interruptions or limited access scenarios.

## Existing Examples / Solutions

At present, a noticeable gap exists in the market for apps that offer communication functionality independent of an internet connection. AndroCom, by enabling text messaging, voice calls and video calls with end-to-end encryption using Raspberry Pi, distinguishes itself as an innovative solution that fills this void. Unlike conventional applications that rely on internet connectivity, AndroCom offers users a novel approach to communication in scenarios where such connectivity may be unavailable or limited, addressing a critical need in today's interconnected world.

## Business Scope

The business scope of AndroCom is promising, offering a unique solution for communication in scenarios with limited or no internet access. It caters to a niche market and educational institutions, presents a valuable tool for disaster recovery and emergency services, and has the potential to serve as a backup communication service during internet outages. With its AD HOC networking capabilities, it can find use in various temporary gathering scenarios. The app's focus on data privacy and security also appeals to users prioritizing secure communication, while its potential global reach ensures a broad user base.

## Useful Tools and Technologies

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

* Kotlin
* Java
* Android Studio
* Figma
* Raspberry Pi
* Python

In our application development, we will employ a hybrid approach, primarily utilizing Kotlin for its modern features and conciseness, while also integrating Java where necessary for specific algorithms, socket programming and several modules. Android Studio will serve as our development environment of choice, offering a comprehensive set of tools for efficient coding and testing. Notably, our app will not rely on an internet connection due to its offline functionality. An AD HOC network is created using Raspberry Pi with a server implemented in Python.

## Project Work Breakdown

The project work breakdown for the AndroCom is given in Figure 1.1.

Figure 1.1: Project Work Breakdown

## Project Timeline

The project timeline for AndroCom is given in Figure 1.2.

Figure 1.2: Project Time Line

# 

# Chapter 2

# Requirement Specification and Analysis

The emphasis of this chapter is on getting an idea of what the requirements are for the intended software. Students who are doing a research related project would provide literature survey for their problems. They are expected to understand the relevant papers and provide summary of the existing work presented in each research paper. Such students should consult their project supervisor for the detailed instructions related to this chapter.

*You should write SRS in precise, clear and plain language so that it can be reviewed by a business analyst or customer representative with minimal technical expertise. However it also contains analytical models (use case diagrams, entity relationship diagrams, data dictionary etc.), which can be used for the detailed design and the development of the software system*.

## Requirement Specification

Requirements specification involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users and documentation of all aspects of the project development process from start to finish. Requirements are a description of how a system should behave or a description of system properties or attributes. It can alternatively be a statement of 'what' an application is expected to do.

## 2.1. Functional Requirements

The Functional Requirements Specification documents the operations and activities that a system must be able to perform. The Functional Requirements Specification is described in such a way that anyone from non-technical audience can understand. Readers should understand the system, but no particular technical knowledge should be required to understand the document.

* Functional Requirements should include:
* Descriptions of data to be entered into the system
* Descriptions of operations performed by each screen
* Descriptions of work-flows performed by the system
* Descriptions of system reports or other outputs
* Who can enter the data into the system
* How the system meets applicable regulatory requirements

**Table 2.1: Functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Functional Requirement** | **Type** | **Status** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |

## 2.2. Non-Functional Requirements

Non-functional requirements cover all the remaining requirements, which are not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviors, for example: “Modified data in a database should be updated for all users accessing it within 2 seconds”. Some typical non-functional requirements include performance, scalability, availability, reliability, maintainability, usability and security.

*You are supposed to discuss both the functional and non-functional requirement for your project in the Part 1 mid term report. Functional requirements describe what the system should do while non-functional requirements describe how the system works. The Format for presenting these requirements is given in Table 2.*

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

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Non Functional Requirements** | **Category** |
| 1 |  |  |
| 2 |  |  |
| 3… |  |  |

## Selected Functional Requirements

List of selected functional requirements for current iteration.

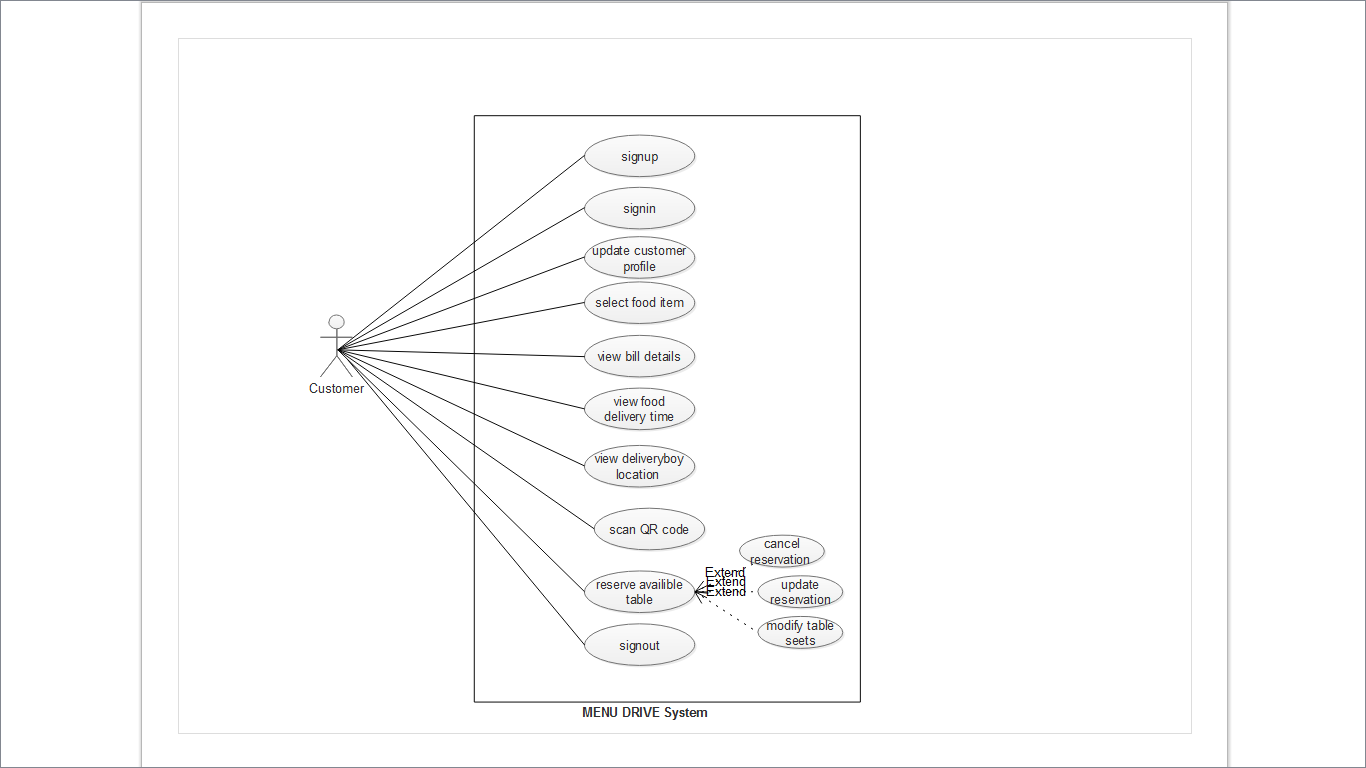
**Table 2.3: Selected Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Functional Requirement** | **Type** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

## System Use Case Modeling

A use case defines a set of use-case instances, where each instance is a sequence of actions a system performs that yields an observable result of value to a particular actor. The functionality of a system is defined by different use cases, each of which represents a specific goal (to obtain the observable result of value) for a particular actor.

*You should develop fully dressed use cases. One way of conceptualize correct use case is by imaging the user interface of all the features of your project. This will help you to improve your project well in time.*

**

**Figure 2.1: Sample Use case Diagram**

**Use Case1 Title:**

*Describe the use case (expected behavior of the software) in the form of steps and sub steps in the format given below. You should also proved a brief description of user interface that will satisfy the requirement of each use case*

**Table 2.1: Use Case 1**

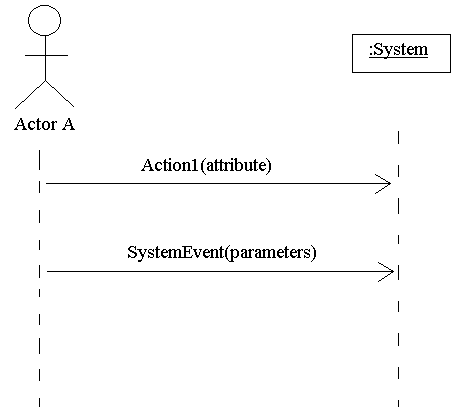
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | [Repeat for multiple use cases] | | | | |
| **Use Case Name:** |  | | | | |
| **Created By:** |  | | **Last Updated By:** | |  |
| **Date Created:** |  | | **Last Revision Date:** | |  |
| **Actors:** | |  | | | |
| **Description:** | |  | | | |
| **Trigger:** | |  | | | |
| **Preconditions:** | |  | | | |
| **Post conditions:** | |  | | | |
| **Normal Flow:** | | Actor | | System | |
|  | |  | |
| **Alternative Flows:** | |  | | | |
| **Exceptions:** | |  | | | |

*Add further use cases in the given format*

## 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.*

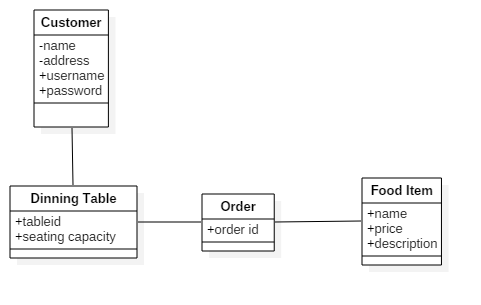


**Figure 2.2: System Sequence Diagram**

## 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.

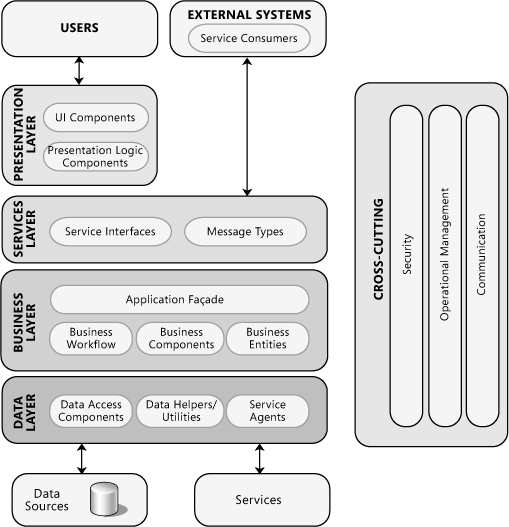
Design pattern is a description or template for how to solve a problem that can be used in many different situations. Object-oriented design patterns typically show relationships and interactions between classes or objects, without specifying the final application classes or objects that are involved.

*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

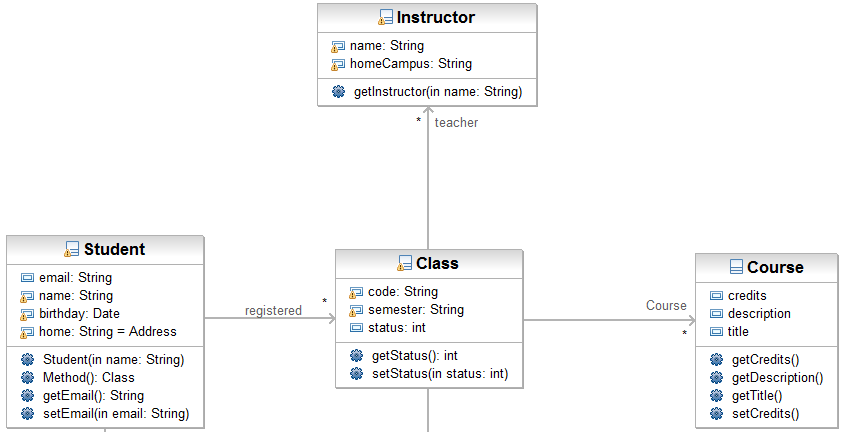


**Figure 3.1: Software Architecture Diagram**

*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.

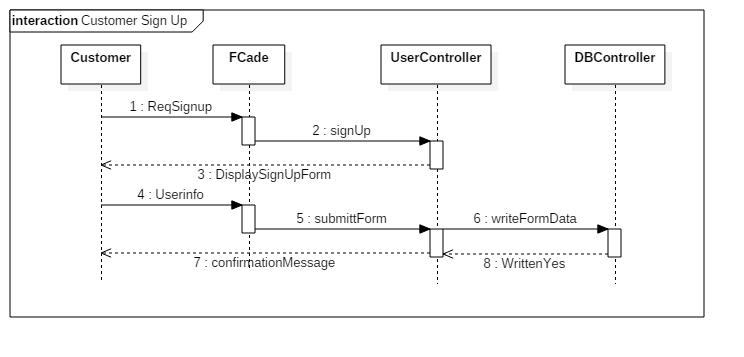


**Figure 3.2: Class Diagram**

## 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.*

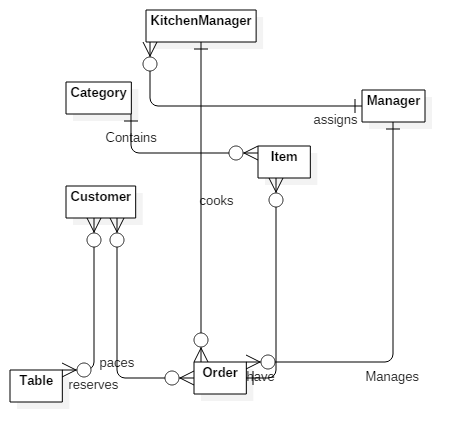


**Figure 3.3: Sequence Diagram**

## 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.*



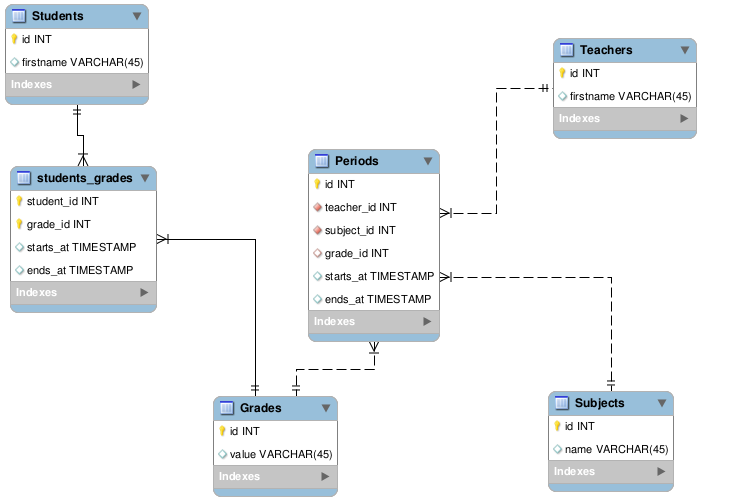
## 

**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.

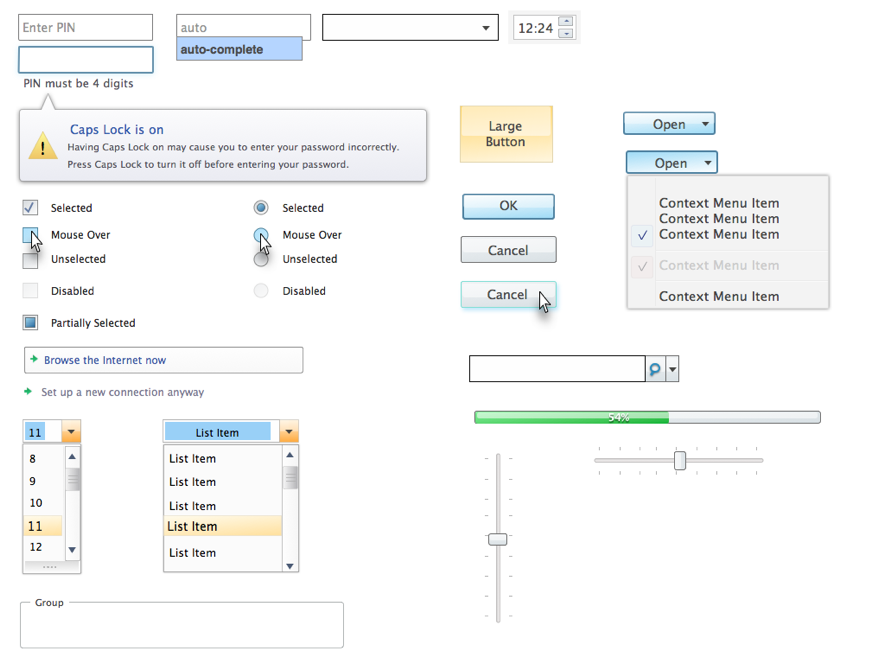
.

**Figure 3.5: Database Schema**

## User Interface Design

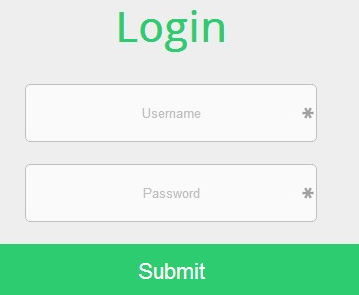
User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. UI brings together concepts from interaction design, visual design, and information architecture.

*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**