**SUPERIOR UNIVERSITY LAHORE**

|  |
| --- |
| Superior Logo |

**Faculty of Computer Science & IT**

**FINAL YEAR PROJECT**

**PROJECT IDEA**

**Colony’s Record Management**

Project ID: **[Issued by FYP Manager]**

**Project Team**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student Name** | **Student ID** | **Program** | **Contact Number** | **Email Address** |
| M.Usman | BCSM-S17-006 | BCS | 03354070136 | bcsm-s17-006@superior.edu.pk |
| Adeel Ali | BCSM-S17-019 | BCS | 03349782003 | bcsm-s17-019@superior.edu.pk |
| M.Usman Tahir | BCSM-S17-014 | BCS | 03056295540 | bcsm-s17-014@superior.edu.pk |

**[Project Supervisor]**

Mr.Waseem Arif

**[** Assistance Professor ]

**Project Report**

**Colony’s Record Management**

**Change Record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author(s)** | **Version** | **Date** | **Notes** | **Supervisor’s Signature** |
|  | 1.0 |  | <Original Draft> |  |
|  |  |  | <Changes Based on Feedback from Supervisor> |  |
|  |  |  | <Changes Based on Feedback From Faculty> |  |
|  |  |  | <Added Project Plan> |  |
|  |  |  | <Changes Based on Feedback from Supervisor> |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**APPROVAL**

|  |  |
| --- | --- |
| **Project Supervisor** | |
| Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |

|  |  |
| --- | --- |
| **Project Manager** | |
| Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
|  |  |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| **Head of the Department** | |
| Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
|  |  |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Dedication

*We dedicate our work of final year project to our respectful teachers, parents and friends, who support our project idea and gave us proper guidance regarding our project, encourage us in every step we took forward in completion of the project. Without their cooperation it would have not been possible to achieve this goal. Specially our teachers who support us and taught us little things that helped us a lot towards our progress. We will always appreciate all of them and remember them in our good books.*

# Acknowledgements

**[“Allah elevates to high positions those from amongst you who are faithful and those who have acquired knowledge “]** (Quran, 58:11)

We really thankful to our FYP Maneger (Zaman Aziz) as well as Supervisor (Waseem Arif), who gave us the golden opportunity to achieve our goal and who gave us continuous guidance, assistance, and inspiration to continue efficiently working on our project and obtain promising results. During this work CRM (Colony’s Record Management) we have learnt a lot of things, do a maximum research and came to know about many new things which will help us in future In-Shaa-Allah.

Last but not least, we would like to present a special thanks to our families, for their love, understanding, encouragement, and confidence in us.

# Executive Summary

We are developing a web-site which holds record of a colony. The web-site will keep the detail record of all the members of the colony, vehicles belonging to members of colony Incoming and outgoing vehicles as well as the visitors to and from the colony with respect to the time. In addition, the website will also have a module as property classified through which the visitors of the website will get the information of property on sale/ rent.

The objective of this project is to develop a web-site in which every member of society should be recorded their entry or leaving time from society and their vehicle’s due to this the security of society would also improve. If anybody wants to complaint against anyone or if they face any kind of problem it helps in investigation by their recorded data. This project will be also useful for the Police Department in case any illegal activity would happens in society they will easily identify the criminals and approached them by their previous history and data. Also, they can use different functionalities of this system like searching properties.

# Table of Contents

[Dedication iv](#_Toc505638416)

[Acknowledgements v](#_Toc505638417)

[Executive Summary vi](#_Toc505638418)

[Table of Contents vii](#_Toc505638419)

[List of Figures ix](#_Toc505638420)

[List of Tables x](#_Toc505638421)

[Chapter 1 1](#_Toc505638422)

[Introduction 1](#_Toc505638423)

[1.1. Background 2](#_Toc505638424)

[1.2. Motivations and Challenges 2](#_Toc505638425)

[1.3. Goals and Objectives 2](#_Toc505638426)

[1.4. Literature Review/Existing Solutions 2](#_Toc505638427)

[1.5. Gap Analysis 2](#_Toc505638428)

[1.6. Proposed Solution 2](#_Toc505638429)

[1.7. Project Plan 3](#_Toc505638430)

[1.7.1. Work Breakdown Structure 3](#_Toc505638431)

[1.7.2. Roles & Responsibility Matrix 3](#_Toc505638432)

[1.7.3. Gantt Chart 3](#_Toc505638433)

[1.8. Report Outline 3](#_Toc505638434)

[Chapter 2 4](#_Toc505638435)

[Software Requirement Specifications 4](#_Toc505638436)

[2.1. Introduction 5](#_Toc505638438)

[2.1.1. Purpose 5](#_Toc505638439)

[2.1.2. Document Conventions 5](#_Toc505638440)

[2.1.3. Intended Audience and Reading Suggestions 5](#_Toc505638441)

[2.1.4. Product Scope 5](#_Toc505638442)

[2.1.5. References 6](#_Toc505638443)

[2.2. Overall Description 6](#_Toc505638444)

[2.2.1. Product Perspective 6](#_Toc505638445)

[2.2.2. Product Functions 6](#_Toc505638446)

[2.2.3. User Classes and Characteristics 6](#_Toc505638447)

[2.2.4. Operating Environment 7](#_Toc505638448)

[2.2.5. Design and Implementation Constraints 7](#_Toc505638449)

[2.2.6. User Documentation 7](#_Toc505638450)

[2.2.7. Assumptions and Dependencies 7](#_Toc505638451)

[2.3. External Interface Requirements 8](#_Toc505638452)

[2.3.1. User Interfaces 8](#_Toc505638453)

[2.3.2. Hardware Interfaces 8](#_Toc505638454)

[2.3.3. Software Interfaces 8](#_Toc505638455)

[2.3.4. Communications Interfaces 9](#_Toc505638456)

[2.4. System Features 9](#_Toc505638457)

[2.4.1. System Feature 1 9](#_Toc505638458)

[2.4.1.1. Description and Priority 9](#_Toc505638459)

[2.4.1.2. Stimulus/Response Sequences 9](#_Toc505638460)

[2.4.1.3. Functional Requirements 9](#_Toc505638461)

[2.4.2. System Feature 2 10](#_Toc505638462)

[2.4.2.1. Description and Priority 10](#_Toc505638463)

[2.4.2.2. Stimulus/Response Sequences 10](#_Toc505638464)

[2.4.2.3. Functional Requirements 10](#_Toc505638465)

[2.4.3. System Feature 3 (and so on) 11](#_Toc505638466)

[2.5. Other Nonfunctional Requirements 11](#_Toc505638467)

[2.5.1. Performance Requirements 11](#_Toc505638468)

[2.5.2. Safety Requirements 11](#_Toc505638469)

[2.5.3. Security Requirements 12](#_Toc505638470)

[2.5.4. Software Quality Attributes 12](#_Toc505638471)

[2.5.5. Business Rules 12](#_Toc505638472)

[2.6. Other Requirements 12](#_Toc505638473)

[Chapter 3 13](#_Toc505638474)

[Use Case Analysis 13](#_Toc505638475)

[3.1. Use Case Model 14](#_Toc505638476)

[3.2. Fully Dressed Use Cases 14](#_Toc505638477)

[Chapter 4 15](#_Toc505638478)

[System Design 15](#_Toc505638479)

[4.1. Architecture Diagram 16](#_Toc505638480)

[4.2. Domain Model 16](#_Toc505638481)

[4.3. Entity Relationship Diagram with data dictionary 16](#_Toc505638482)

[4.4. Class Diagram 17](#_Toc505638483)

[4.5. Sequence / Collaboration Diagram 17](#_Toc505638484)

[4.6. Operation contracts 17](#_Toc505638485)

[4.7. Activity Diagram 18](#_Toc505638486)

[4.8. State Transition Diagram 18](#_Toc505638487)

[4.9. Component Diagram 18](#_Toc505638488)

[4.10. Deployment Diagram 19](#_Toc505638489)

[4.11. Data Flow diagram [only if structured approach is used - Level 0 and 1] 19](#_Toc505638490)

[Chapter 5 20](#_Toc505638491)

[Implementation 20](#_Toc505638492)

[5.1. Important Flow Control/Pseudo codes 21](#_Toc505638493)

[5.2. Components, Libraries, Web Services and stubs 21](#_Toc505638494)

[5.3. Deployment Environment 21](#_Toc505638495)

[5.4. Tools and Techniques 22](#_Toc505638496)

[5.5. Best Practices / Coding Standards 22](#_Toc505638497)

[5.6. Version Control 22](#_Toc505638498)

[Appendices 23](#_Toc505638499)

[Appendix A: Information / Promotional Material 24](#_Toc505638500)

[Reference and Bibliography 27](#_Toc505638502)

[Index 29](#_Toc505638503)

# 

# List of Figures

1.1 Caption of first figure of first chapter 6

1.2 Caption of second figure of first chapter 7

2.1 Caption of first figure of second chapter 14

2.2 Caption of second figure of second chapter 22

2.3 Caption of third figure of second chapter 26

5.1 Caption of first figure of fifth chapter 49

5.2 Caption of second figure of fifth chapter 49

# 

# List of Tables

1.1 label of first table of first chapter 6

1.2 label of second table of first chapter 7

2.1 label of first table of second chapter 14

2.2 label of second table of second chapter 22

2.3 label of third table of second chapter 26

5.1 label of first table of fifth chapter 49

5.2 label of second table of fifth chapter 49

# Chapter 1

# Introduction

# Description: C:\Users\Mughal usman\Desktop\FYP Doc\Requirements-Gathering.jpg

**Chapter 1:**

# Executive Summary

We are developing a web-site which holds record of a colony. The web-site will keep the detail record of all the members of the colony, vehicles belonging to members of colony Incoming and outgoing vehicles as well as the visitors to and from the colony with respect to the time. In addition, the website will also have a module as property classified through which the visitors of the website will get the information of property on sale/ rent.

The objective of this project is to develop a web-site in which every member of society should be recorded their entry or leaving time from society and their vehicle’s due to this the security of society would also improve. If anybody wants to complaint against anyone or if they face any kind of problem it helps in investigation by their recorded data. This project will be also useful for the Police Department in case any illegal activity would happens in society they will easily identify the criminals and approached them by their previous history and data. Also, they can use different functionalities of this system like searching properties.

**1-Introduction**

The main aim of this project is to develop a Responsive web-based system of record management of a colony. The website will keep the record of all the members of the colony, vehicles belonging to members of colony, incoming and outgoing vehicles as well as the visitors to and from the colony with respect to the time. In addition, the website will also have a module as property classified through which the visitors of the website will get the information of property on sale/ rent.

## Background

We are building up system which holds record of a settlement. The system will keep the detail record of the considerable number of individuals from the state, vehicles having a place with individuals from settlement Incoming and cordial vehicles just as the guests to and from the province as for the time. Likewise, the site will likewise have a module as property ordered through which the guests of the site will get the data of property marked down/lease.

The goal of this task is to build up a site in which each citizen ought to be recorded their entrance or leaving time from society and their vehicles because of this the security of society would likewise improve. On the off chance that anyone needs to grumbling against anybody or on the off chance that they face any sort of issue its aides in examination by their recorded data. This task will be additionally valuable for the Police Department on the off chance that any criminal behavior would occurs in the public eye they will effectively distinguish the crooks and moved toward them by their past history and information. Additionally, they can utilize various functionalities of this framework like looking through properties.

The pace of criminal episodes and a ton of other criminal actuates are expanding day by day. It’s getting hard to keep up the security perceived the genuine evildoer. The framework of society nowadays is worked physically that isn't fruitful the information is changed effectively despite the fact that it extremely hard to keep up past information records physically. This framework would assist with beating these issues.

## Motivations and Challenges

* Evaluation criteria of term and final year project do not match with industrial requirements.
* Dataset of term and final year projects is not available in the database to build a model.
* No optimal system for such criteria is present.
* Manual way of record should be online.
* No proper record was available for Residential/ visitors.
* Manual way for entering in the colony.
* Lack of security due to outsiders.
* No uses of handwritten notebooks or spread-sheets.

## Goals and Objectives

The goals of High-Quality **Colony’s Record Management** are to: hold information about a particular colony and improve the security system moreover facilitate the people about property affairs. This project may help decries the size of crime in society it also has a plus point that recognized the outsider and residential.

* Collect the information of colony’s members, outsiders and manage them
* Prevented from the manual system (afraid of data loss)
* Would not be worried about any deal related to the colony
* Strict security system overcome the crime rate in the society
* Make more secure

## Literature Review/Existing Solutions

Some existing tools like Microsoft Excel and colony management cloud-based tools are either cumbersome or expensive. There is a need for inexpensive and uncomplicated colony management software with specific individual database solutions also there is no searching facility about the colony rent sale purchased in these systems. More over if anyone want to buy property, he has to visit property dealer for related a lot of time. If we are doing the system manually, so many minor errors will occur. Error detection in the previous entries made and data cross verification is another important function. These are done manually, and it would take time. The existing system has more workload for the authorized person. We have developed a database application named **Colony management system** to meet most of the requirements of such manual colony management programs.

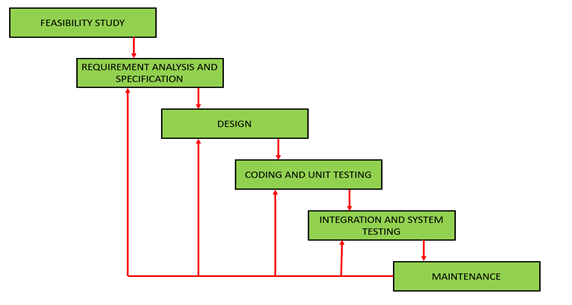
## Gap Analysis

Colony’s Record Management will help to kept the record of colony’s members and outsiders. This will overcome the loss of data and maintain security system even it also helps to search out the results like buying/selling according to your desire. At current state we are working on data management further in future we will work on data management system with finger print identification.

## Proposed Solution

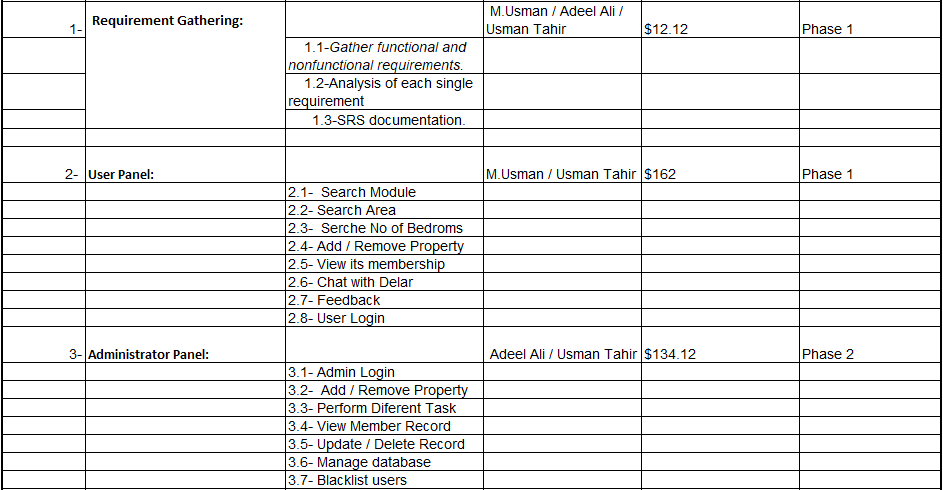
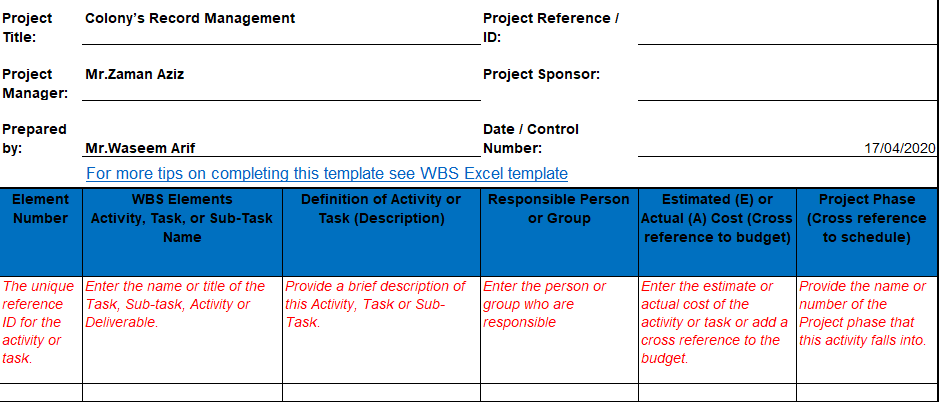
* We will provide evaluation criteria for every level like Entry time/ leaving time /Stay area (block/house no etc.)
* Dataset of all record’s will available
* Lack of security for data
* Easily searching Rent/Sale property
* Add visitor record and create visitor card according to relevance information

## Project Plan

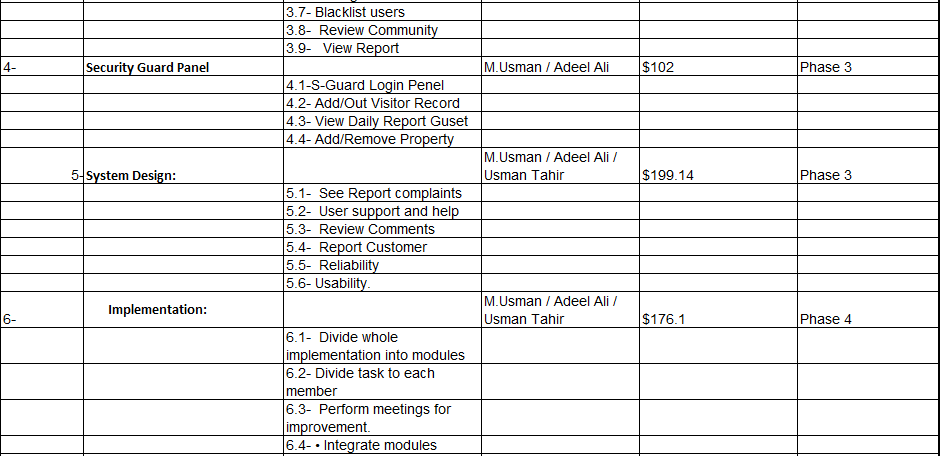


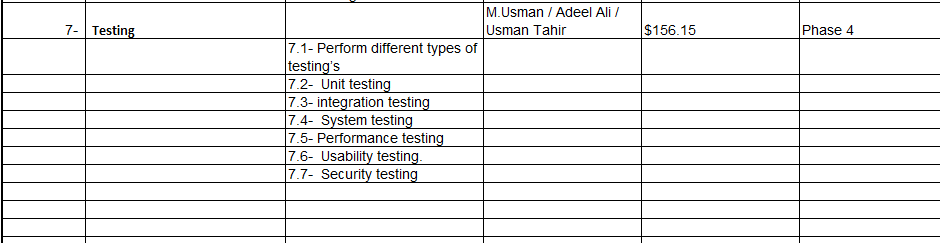
The project will be managed to divide into modules a module to clearly understand how we take next step with the specified time. First we use to gather the requirement from manager and then do all the phases of Software making like after requirement we analyze them and make design of our gathered requirements and divide the modules to our members then do implementation and later we do integration of the modules and last we do test our application and then make it available for our customers. We follow the agile methodology in our project and work on each of single iteration within the time specified. We are using this methodology for Improvement in our project and can adopt change easily in each of single iteration.

## Work Breakdown Structure:

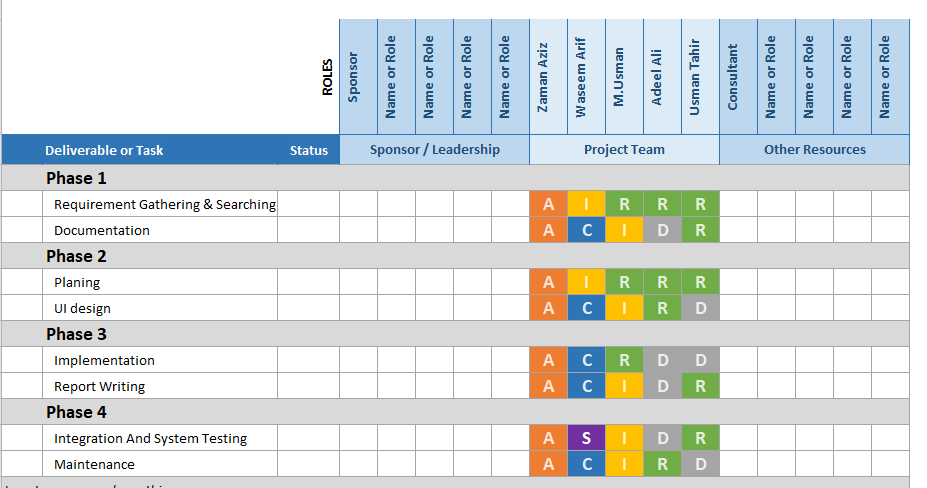


## Work Breakdown Structure:

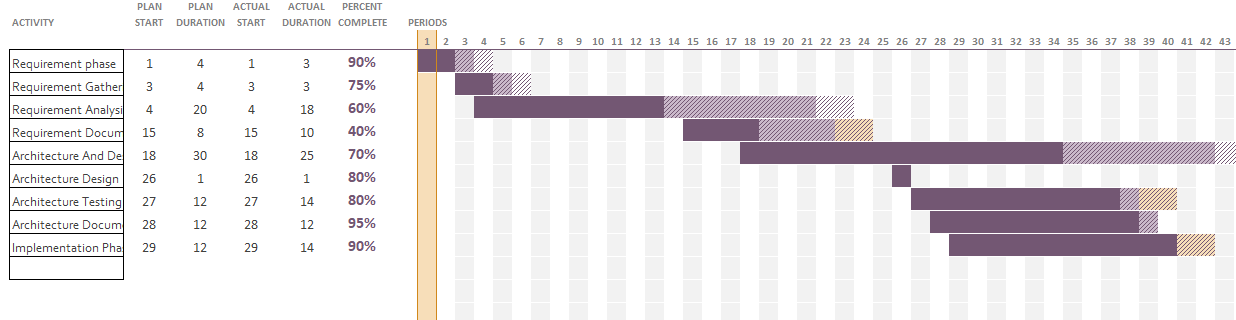




## Roles & Responsibility Matrix:



## Gantt Chart:



* 1. **Report Outline:**

The main Report of this project is to develop an Online Colony’s Record Management system which is easily accessible to the people. Report outline following that:

* All the system inputs are validated and don’t accept the invalid number or commands.
* Save data smoothly
* With the help of data approaches to the suspect of any incident.
* It generates the pure and clear history of all the related colony members and outsiders.
* All activities track under the rules of colony

**---------------------------------------------ITRATION#1\_COMPLETE-------------------------------------------------**

# Chapter 2

# Software Requirement Specifications

**Chapter 2:** Software Requirement Specifications



## Introduction

## Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

## 

## Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

## 

## Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

## Overall Description

## Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

## Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

## External Interface Requirements

## 

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## 

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

## System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

## Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

## Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

## Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-SF1-1: <Write your requirement here>

REQ-SF1-2:

REQ-SF1-3:

## System Feature 2

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

## Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

## Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

## Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-SF2-1:

REQ-SF2-2:

REQ-SF2-3:

## System Feature 3 (and so on)

## Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

## Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

# Chapter 3

# Use Case Analysis

**Chapter 3:** System Analysis

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

[*Between 4 to 8 lines describe what is this chapter all about*]

## Use Case Model

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Fully Dressed Use Cases

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

# Chapter 4

# System Design

**Chapter 4:** System Design

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

[*Between 4 to 8 lines describe what is this chapter all about*]

## Architecture Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Domain Model

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Entity Relationship Diagram with data dictionary

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Class Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Sequence / Collaboration Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Operation contracts

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Activity Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## State Transition Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Component Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Deployment Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Data Flow diagram [*only if structured approach is used - Level 0 and 1*]

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

# Chapter 5

# Implementation

**Chapter 5:** Implementation

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

[*Between 4 to 8 lines describe what is this chapter all about*]

## Important Flow Control/Pseudo codes

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Components, Libraries, Web Services and stubs

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Deployment Environment

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Tools and Techniques

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Best Practices / Coding Standards

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Version Control

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

# Appendices

# Appendix A: Information / Promotional Material

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

[*Between 4 to 8 lines describe what is this appendix all about*]

* 1. **Broacher**

* 1. **Flyer**
  2. **Standee**
  3. **Banner**
  4. **First Level heading [16 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

* + 1. **Second level heading [14 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

* + - 1. **Third level heading [12 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

# Appendix [no.]: Appendix Title

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

[*Between 4 to 8 lines describe what is this chapter all about*]

* 1. **First Level heading [16 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

* + 1. **Second level heading [14 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

* + - 1. **Third level heading [12 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

# Reference and Bibliography

**Reference and Bibliography**

[1] M. Sher, M. Rehman, “*Title of the Paper*” Conference name/Journal Name, Edition, Volume, Issue, ISBN/ISSN, PP, Publisher/City-Country, Year.

[2] ……

# Index

**Index**

**[A]**

**[B]**

**[C]**