Software Requirements Specification

Version 2.0

Real-time Cupboard Design

Important Note: This template has been adapted from the SRS Template by Karl E. Wiegers which is available for download at http://www.processimpact.com/

Team 2

Member Name	Member Roll #	Primary Responsibility
M.Mujahid Iqbal	15L-4105	Project Manager
Asim Fayaz	15L-4097	IV & V Person
Ahmad Raza	15L-4041	Functional
Timur Asif	15L-4012	Non-Functional Requirements
Hassan Jalal	14L- 4344	Risk Analysis and Product Description

Important Note: This template has been adapted from the SRS Template by Karl E. Wiegers which is

Table of Contents

Important Note: This template has been adapted from the SRS Template by Karl E. Wiegers which is

Revision History

Name(s)	Date	Reason(s) For Change(s)	Version

Important Note: This template has been adapted from the SRS Template by Karl E. Wiegers which is

1. Introduction

Product

A desktop based application for users to design a cupboard design in real time. The product supports both the internal and external design of a cupboard. Drawers and portions or shelves are included in interior and knobs are included in exterior design. Users can customize the design by moving portions of cupboard according to the requirements. The application also gives the estimated cost for the design as well as allows the placement of order directly from application.

Scope

The scope of the project is only limited to one manufacturer i.e. clients can make designs and send an email of those images to the manufacturer who will then actually make cupboards accordingly and will deliver back.

Business Goals

Business goals includes giving relaxation to the clients so that they don't have to always come to the shop and have a proper conversation with the manufacturer but can make or choose designs while sitting in home. Also, since customers are having a portable system of getting cupboards ready so they will prefer this desktop application manufacturer over the ones where they have to visit and tell every detail in person so business of manufacturer having this application will grow.

Document Conventions

- SM stands for Senior Manager.
- DB stands for Database.
- Citation format is APA.
- All heading are in bold.

References

1. Bandakkanavar, R. (2018). Software Requirements Specification document with example - Krazytech. Retrieved from https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

[1] "Getting started with Javafx" vol. 8, August 1, 2016. [Online]. docs.oracle.com/javase/8/javafx

[2] Kaul, Jeet, "JavaFX — the road ahead", December 18, 2008. [Online]. https://web.archive.org/web/20081217162601/http://blogs.sun.com/meetjeet/entry/javafx_the_road_ahead

Important Note: This template has been adapted from the SRS Template by Karl E. Wiegers which is

2. Overall Description

Product Features

User will download this application and then make design of cupboard. After designing the whole cupboard he/she will set budget and delivery time. Then after he is done, the application will send an email to the manufacturer with all images of cupboard, delivery date and budget of the customer. The manufacturer will then respond in email to the customer whether he has accepted his order, rejected or is at stall. Once he has accepted the order and delivered it, the order will be stored in the personal database (DB) of the user's application. So that he can have records of all the orders he has placed till date.

User Classes and Characteristics

There will be just one user i.e. customer because manufacturer will just receive an email from which he has to decide whether to accept it or reject it or put it on hold. So only customer will use this application.

Customer

There will be no sign-up option because manufacturer doesn't care who is his customer, so any customer who has this application will open the application and start making cupboard designs, can then send email and will receive response in the email.

Operating Environment

Hardware requirements

A working PC, laptop and a working internet.

Software Environment

JRE(Java Runtime Environment)

No software requirements and operating system version restrictions.

Design and Implementation Constraints

Hardware limitations: Since application is desktop based so customer should only be able to run it on his laptop or computer.

Interfaces to other Applications: Just email will be sent i.e. email feature that will be used will use Google's Gmail API so a working Gmail account is mandatory.

Language and Communication Protocols are that language used throughout should be English with technical and precise wording and measurements will b made in inches or in meters.

Important Note: This template has been adapted from the SRS Template by Karl E. Wiegers which is

available for download at [illegible]

Security Considerations are that he email will be open to any kind of phishing or hacking but responsible users can avoid that by making contact to the manufacturer so they can confirm their order or can discard one if someone placed an order using their email.

Design Conventions: The desktop application's graphical user interface is as simple and meaningful as possible so any customer can use it easily.

Programming Standards: Java Fx and swing library was used along with java's canvas API to draw objects.

Assumptions and Dependencies

The assumptions made are that:-

- *Each user will use it just for meaningful purpose and won't give other people's emails to create confusion and false orders.*
- *Each user will have a working e-mail id where they can get response regarding their order i.e. the application will not handle the response.*
- *There is only one manufacturer, so all orders will go to just one fixed person whose email will be predefined in the application.*
- *Since response will be given in email so the entire user's orders will be added to db instantly, this will be stored locally because there is no other way for the application to keep track of customer's accepted orders.*

3. Functional Requirements

3.1:

Identifier	Design Cupboard	
Purpose	System shall provide a mechanism to draw cupboard design	
Priority	High	
Actors	User	
Pre-condition(s)	Application is running	
Post-condition(s)	Complete design is displayed	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User draws the cupboard design	System show the design
2	User clicks on save design	Prompts for the design name
3	Enters the design name to be saved	Displays "Design Saved Successfully" message
Alternate Course of Action 1 (Format is not Correct)		
S#	**Actor Action**	**System Response**
1	User clicks on edit the design	System shows saved designs
2	User selects one of the designs	System opens the saved design and display it.
Go to line no 2 in Typical course		

3.2

Identifier	Analyze Payment
Purpose	System shall provide means to allow the user to see the estimated cost
Priority	High
Actors	User
Pre-condition(s)	Design is built and saved
Post-condition(s)	The final cost is displayed for the design

Typical Course of Action

S#	Actor Action	System Response
1	User clicks on Payment Analysis tab	System opens the amount details window
2		System shows the payment details
3	User views the details of cost and proceed	System shows details in table format with scrolling

3.3

Identifier	Place Order
Purpose	System shall provide method to allow the placement of order of their custom design
Priority	High
Actors	User and System
Pre-condition(s)	Design is built and saved
Post-condition(s)	Order is placed and now further correspondence will be done with manufacturer directly.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User clicks on Payment Analysis tab	System opens the amount details window
2		System shows the payment details
3	User clicks on order now button	Prompts for details
4	User enters personal details required	System proceeds with entered inputs
5	User clicks on submit	System sends all the entered details along with pictures of the design to the manufacturer via Email
Alternate Course of Action 4 (Format is not Correct)		
1	User clicks on order now button	System shows "Please enter correct email" message
2	User enters correct credentials	System proceeds accordingly

3.4

Identifier	View Placed Orders	
Purpose	System shall provide method to display the orders customer has placed	
Priority	Medium	
Actors	User	
Pre-condition(s)	Design is built and Order is placed	
Post-condition(s)	Placed orders cannot be reversed from this method	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User clicks on View Placed Orders tab	System opens the placed orders window
2		System loads placed order data from database and display
Alternate Course of Action 1 (Format is not Correct)		
1	User clicks on View Placed Orders tab	System opens the placed orders window
2		System displays "No order Placed Yet"

4. Nonfunctional Requirements

Performance Requirements

- System shall send order form to Manufacturer within 1 minute given that user has stable internet connection.
- Upon each addition in the design, system should create and display the component just upon clicked.
- While saving order to database and during email sending, system should not stuck user in the interface due to I/O operations etc.

Usability

- Easy to use interface for experts as well as novice users.
- Reversible actions allowed for customization of design and prevention of mistakes

Security Requirements

Authentication

The user must have a valid email address in order for him/her to send or place order to the manufacturer. The email address will be validated according to some regular expressions used for validation. Moreover, exception will be generated by email API when wrong input of email is entered. Also the database should be locally created (using windows authentication) rather than on any other server to make sure orders are for the specific customer only.

5. Other Requirements

Database used will be MySQL database for allowing clients to store their order locally. Database will only contain information of date in which order was placed, dimensions of the cupboard and total price for the design. Besides, the users of this software must be familiar with basic cupboard designing terms like shelves, drawers, knobs etc.

Appendix A: Glossary

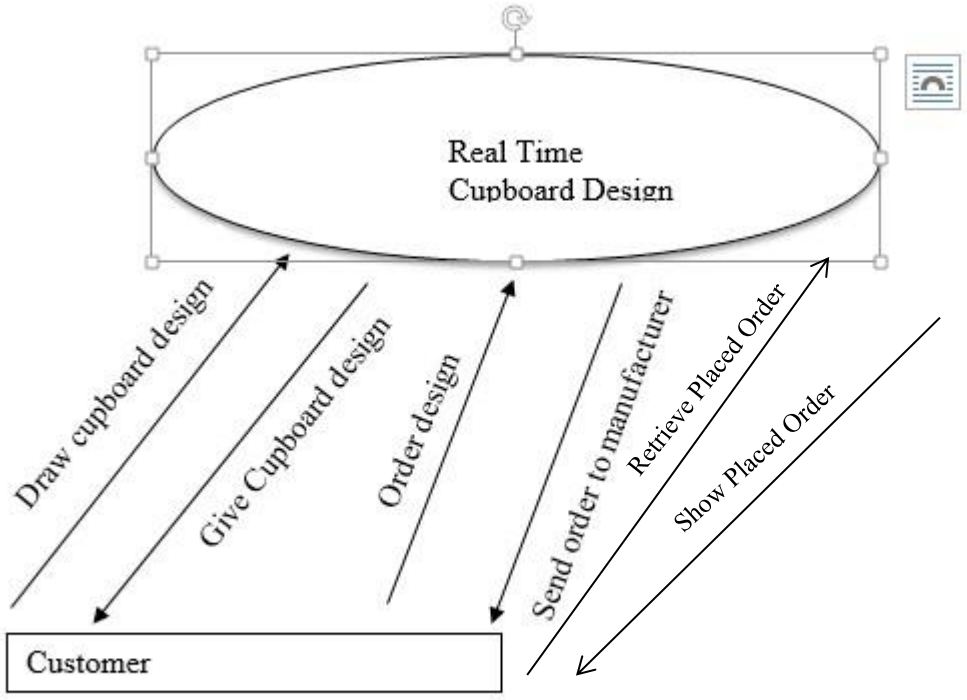
JavaFX

JavaFX [1] is a set of graphics and media packages that enable developers to design, create, test, debug, and deploy rich client applications that operate consistently across diverse platforms. JavaFX have FXML and Scene builder, built-in UI controls and CSS and Canvas API for creating applications. The cross-platform compatibility enables a consistent runtime experience for JavaFX applications, developers and users. Due to this portability it can even be run on other computers (MAC) [2].

Appendix B: Analysis Models

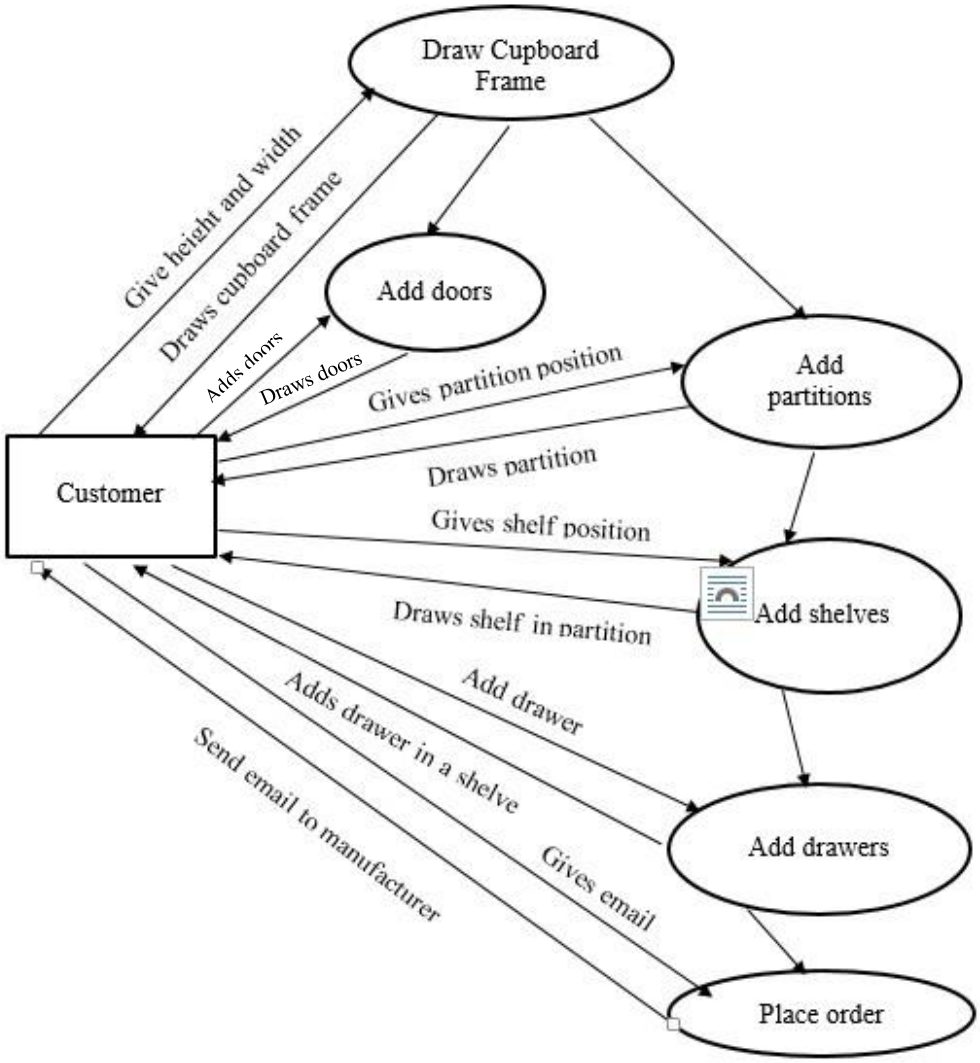
Level 0 DFD:

In level 0 DFD the basic overview of the application is shown. User gives the cupboard design and places the order and manufacturer then contacts him and delivers the order made. Payment method is not in the scope of this project, because it is like OLX app which just connects the buyer (client) and seller(manufacturer) and they can then exchange contact details and work something out then.



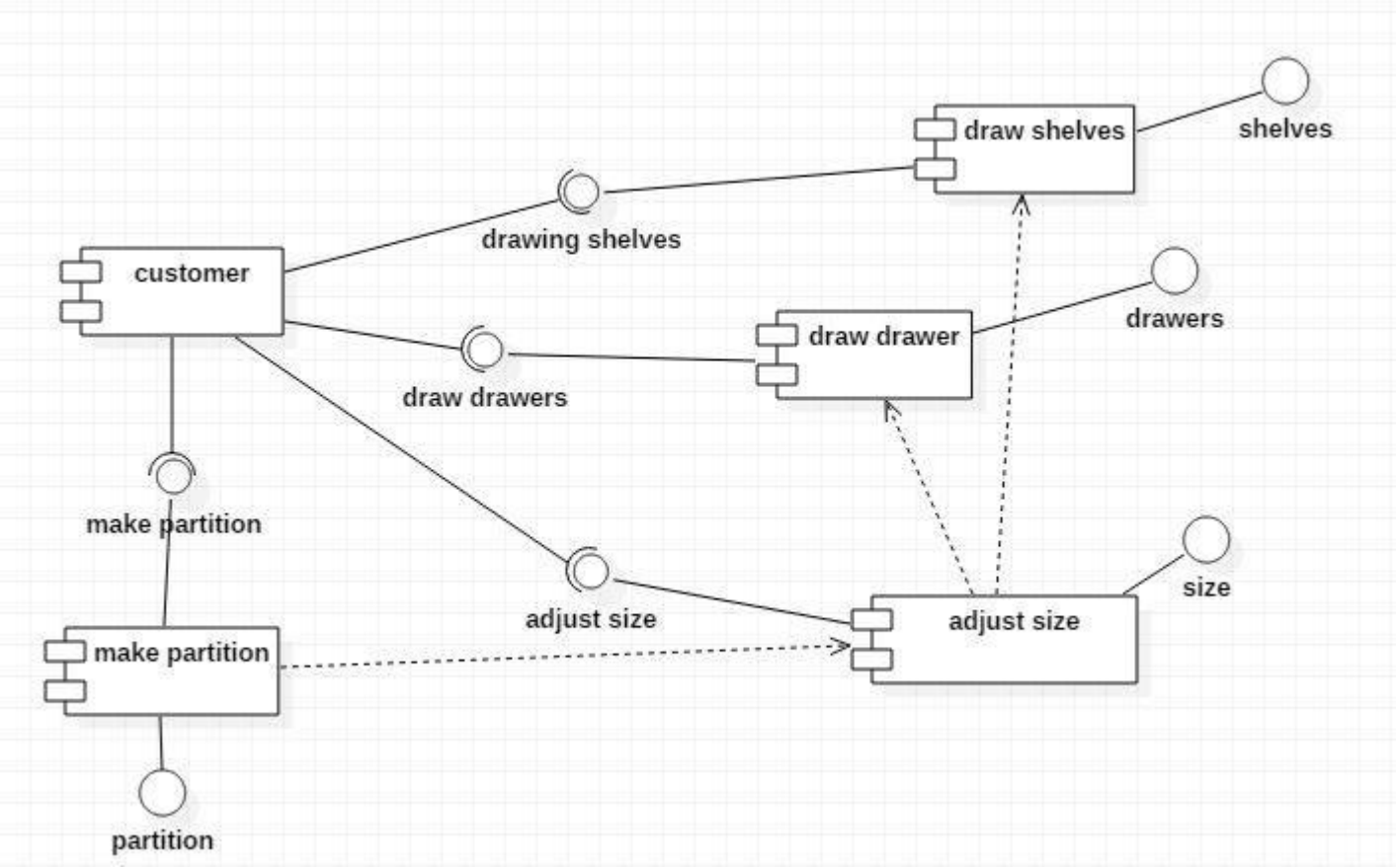
Level 1 DFD:

This is level 1 DFD which elaborates in detail the function being done by our application. Customer can add doors, partitions, shelves, drawers and can specify their sizes and can see the design and changes in real time instantly. Also order can be placed and can then be sent over by GMAIL to the manufacturer.

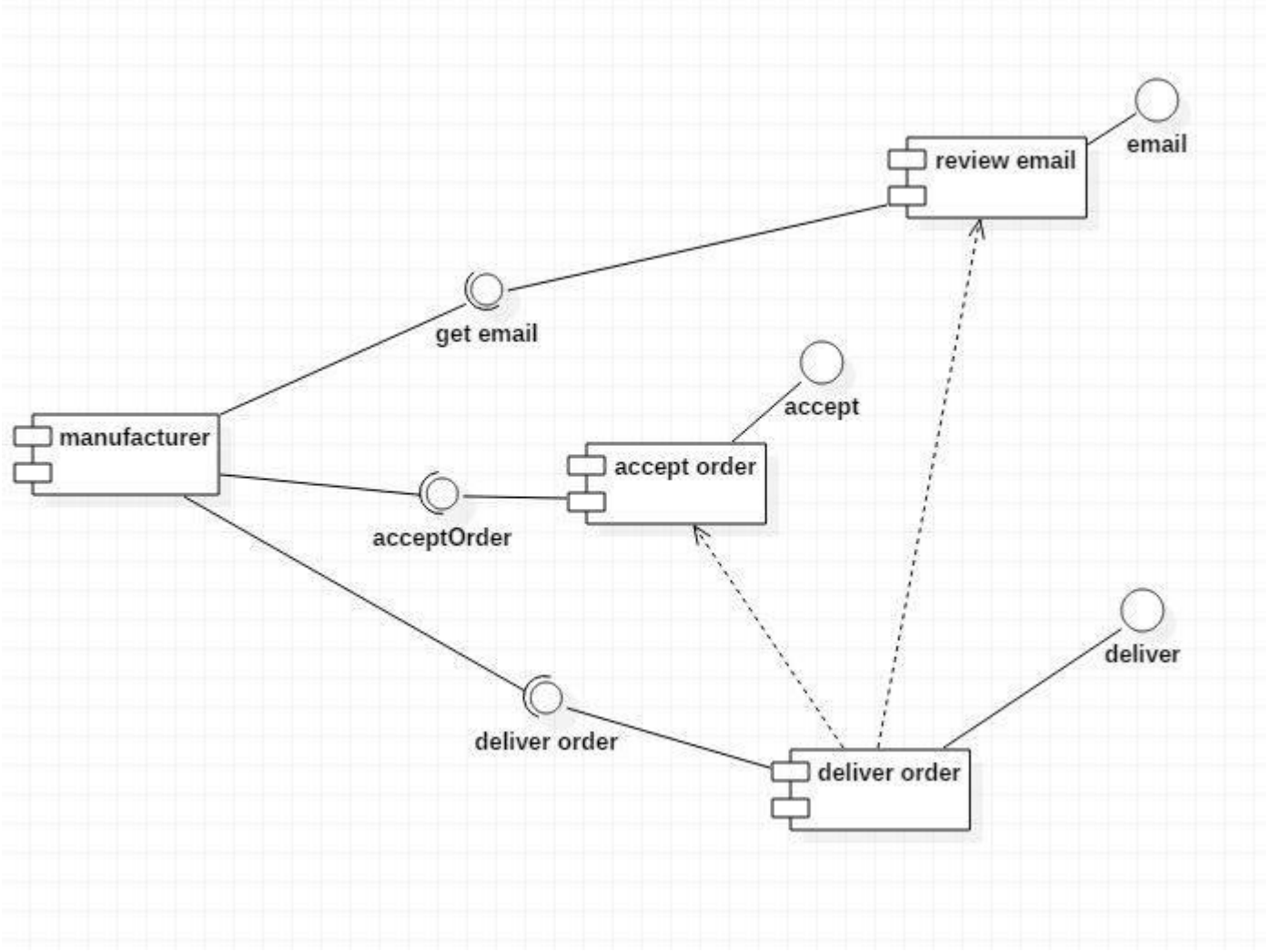


Appendix C: Design Models

Component Diagrams



Important Note: This template has been adapted from the SRS Template by Karl E. Wiegers which is



Important Note: This template has been adapted from the SRS Template by Karl E. Wiegers which is

Appendix F: IV & V Report

IV & V Resource

Name Roll # Signature

S#	Defect Description	Origin Stage	Status	Fix Time	
				Hours	Minutes
1	Spelling mistakes throughout	Phase 2	Fixed		30
2	Use case diagram had missing sections	Phase 1	Fixed	1	0
3	Nonfunctional requirements had ambiguity	Phase 2	Fixed	1	0

Table 3: List of non-trivial defects

Important Note: This template has been adapted from the SRS Template by Karl E. Wiegers which is available for download at [illegible]

Appendix G: Risk Report

[1]Project Risks

Risk Description	Impact (1 – 10)	Probability (0 – 1)	[2]Risk Exposure	Weeks Active	Mitigation Strategy
Time Management risk	10	0.9	9.0	8	assigning deadlines to every team member and ever module of project
work schedule risk	8	0.8	6.4	1	Pre plan the flow of tasks. decide which tasks could be performed in parallel and which are dependent on other tasks.
Quality maintenance risk	7	0.5	3.5	8	checking the quality of every document and module in parallel with development
Design risk	5	0.35	1.75	3	meeting the client and discuss the design and proceed accordingly

Usability risk	5	0.3	1.5	1	designer will work to make user friendly design which is both efficient and easy to interact with
integration risk	4	2.5	1	8	following standard conventions to make module integration easy

1 Risks should be sorted in descending order of risk exposure.

2 Risk Exposure = Risk Impact x Risk Probability

Appendix H: Activity Timesheet

Activity	Time	
	Hours	**Minutes**
Requirements Engineering	14	0
Analysis and Design	24	
Implementation	106	
Testing		
Deployment		
Project Management		
IV & V	2	30

Project Manager

Muhammad Mujahid 15L-4105

Name Roll # Signature

Important Note: This template has been adapted from the SRS Template by Karl E. Wiegers which is

Important Note: This template has been adapted from the SRS Template by Karl E. Wiegers which is