Software Requirements Specification

Version 1.0

January 17, 2018

Student Manager System

Arora Isha Hemant 16BCE2003
Sagarika Sardesai 16BCE2189
Liza Stephen 16BCE2251

Submitted in partial fulfillment
Of the requirements of
CSE 3001 Software Engineering

CONTENTS

1.0. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the Student Manager System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2. Scope of Project

This software system will be a Student Manager System for a student of an educational institute. This system will be designed to maximize the student's productivity by providing tools to assist in managing academics and health, which would otherwise have to be performed manually, which would have consumed time, money and efforts of the institute officials and the student too. By maximizing the usability the system will meet the student's needs while remaining easy to understand and use.

More specifically, this system is designed to allow a student to review his/her academic and health choices. The software will facilitate communication between faculty members and the students via a message portal. It will also provide for booking online specialized appointments with the health care centre. The system also contains tools to calculate student attendance and analyze student performance in exams.

1.3. Abbreviations and Definitions

Term	Definition
Professor	Person who will update student marks and attendance. They will also decide the field in which they want to do research.
Database	Collection of all the information monitored by this system.
Student	Person who applies for medical appointments, applies for research under professors.
Field	The area of interest for research.
Health Centre	The clinic that will provide medical services.
Health Guardian	The person that will confirm the specialized appointments based upon availability of staff and resources.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Stakeholder	Any person with an interest in the project who is not a developer.

Abbreviations:

AC- Attendance Calculator
HC- Health Centre
HG- Health Guardian
SC- Student Calendar
SRS- Software Requirement Specification

1.4. References

www.lucid.com

www.smartsheet.com

1.5. Overview of Document

- The SRS will provide a detailed description of the Student Manager System. This document will provide the outline of the requirements, overview of the characteristics and constraints of the system.
- The SRS will provide the general factors that affect the product and its requirements.
- It provides the background for those requirements.
- The items such as product perspective, product function, user characteristics, constraints, assumptions and dependencies and requirements subsets are described in this section.
- The SRS contains all the software requirements mentioned in detail sufficient enough to enable designers to design the system to satisfy the requirements and testers to test if the system satisfies those requirements.

2.0. Overall Description

2.1. Product Description

This is a system allows the student to manage tasks that would have consumed time and money otherwise, online. It can be used by the student to calculate their attendance in their respective courses. The student can also view their academic progress in the Performance Analysis feature, that will enable the student to choose courses that'll boost their performance, based on their current performance. They can also, book appointments for blood tests or specialized doctors online. These appointments will be confirmed by the HG based on the availability of resources and doctors. The student can also take up research projects under professors, who will list their field of interest in the research area. The professors will have to list the number of students they are willing to take and the students will apply accordingly, based on availability of seats.

2.2. Product Function

- The user has to register with his/her name and id (professor or student) to access this system.
- They have to select the medical department and time from the given list, for a medical appointment at the health centre.
- The student has to choose faculty for research project based on availability of seats.
- The professor will upload student marks in a respective subject. This will be used in calculating the student performance.
- The professor will also upload the attendance, which will allow the student to review their attendance.

- Login Capabilities: User can login using his username and password.
- Mobile devices: This system should be supported if a user logs in using mobile devices with internet.

2.3. User Characteristics

- Type (i.e. the user could either be a student or a professor)
- If user is professor, activity is limited to project and research integration through the online public forum and following up with personal messaging
- If user is student,
  - Registration ID (Primary key)
  - Personal information
  - Major course
  - Minor course (if applicable)
  - Ratio of credits completed to total credits in course
  - Grades
  - Subjects currently undertaken and attendance in each
  - Projects and research completed and/or published
  - Physical examination report

2.4. Constraints

- Username and password constraints
- Security constraints (CAPTCHA)
- Response time constraints
- Cost constraints for integration

2.5. Assumptions & Dependencies

- User has basic knowledge of computers.
- User should have sufficient knowledge of English since the system interface will be in English.
- The system is fast.
- Internet connection is available to all the users who use this subsystem.
- The user is assumed to give system correct information about his details.
- The system will have simple and easy to use interfaces.
- All the necessary information is present in the database.
- Provides accurate data.

3.0. Specific Requirements

3.1. Functional Requirements

3.1.1.Student Calendar: This feature will allow the student to view the upcoming deadlines for the current month in a calendar form. The deadline days will be highlighted and on clicking on them, student will be able to view the list of courses for which that deadline exists. The academic calendar will be incorporated in this as well. This is the first thing that the student will see after logging in.

3.1.2.Health Care: The student will be able to access his/her medical history with Chettinad hospital. They'll be able to book appointments or even renew prescriptions.

3.1.3.Performance Analysis: The student performance will be analyzed. This will be shown in the form of graphs in the categories UC, UE, PC, PE. This performance will further suggest which courses the student should take up in order to improve grades or CGPA.

3.1.4.Attendance Calculator: The class attendance will be calculated after every class. The student will be able to predict how many classes they can miss or even how many classes they must attend to maintain the minimum requirement.

3.1.5.Integrated Projects And Research: If a student is interested in a particular field for a project or a research paper, he can update the status here. Similarly, the faculty will also list his interests here. Based on the matches, the student will be shown a list of professors interested in that field and similarly the teachers will be able to do so as well. They can then choose their preference.

4.0. Non-Functional Requirements

4.1. Performance Requirements

- Throughput: The throughput should be as high as possible. We should be able to attain maximum output in minimum time.
- Resource Utilization: Resources are modified according to user requirements.

4.2. Requirement of Accuracy

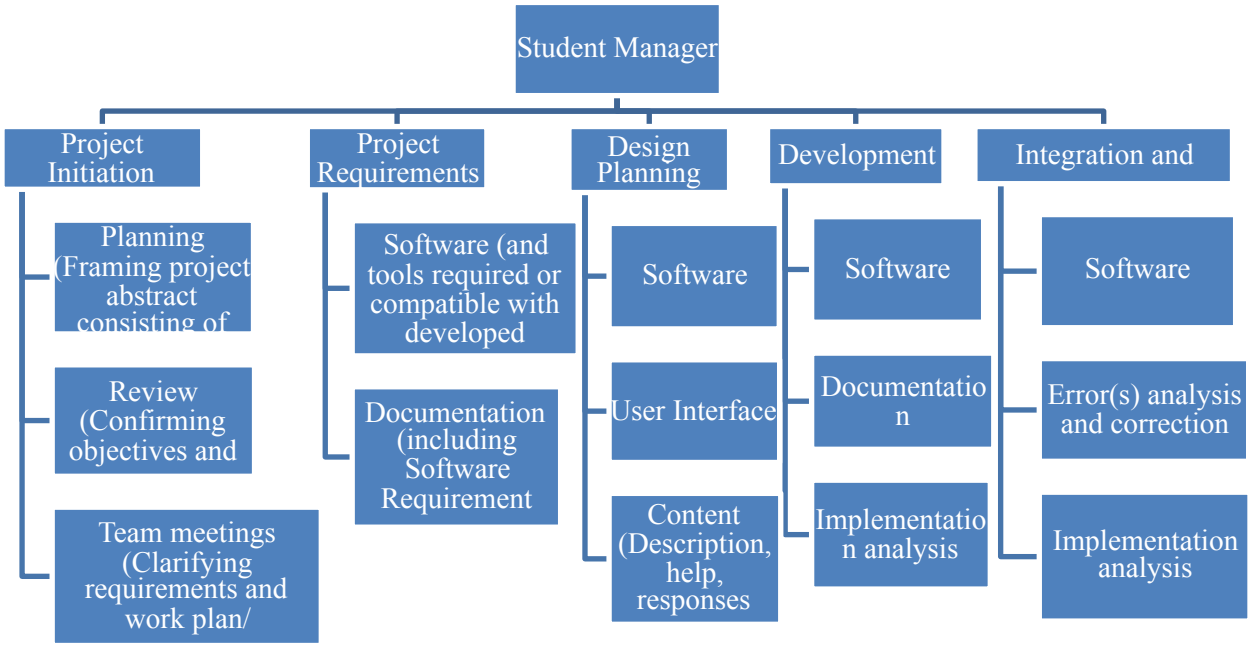
- The information provided in the database and by the user should be correct, completely authentic and should match.

4.3. Usability Requirements

- The interface is simple and easy to use.
- System is user friendly and self-explanatory.
- This system can be used by the students, the faculty as well as the administration.

5.0. Components Asked

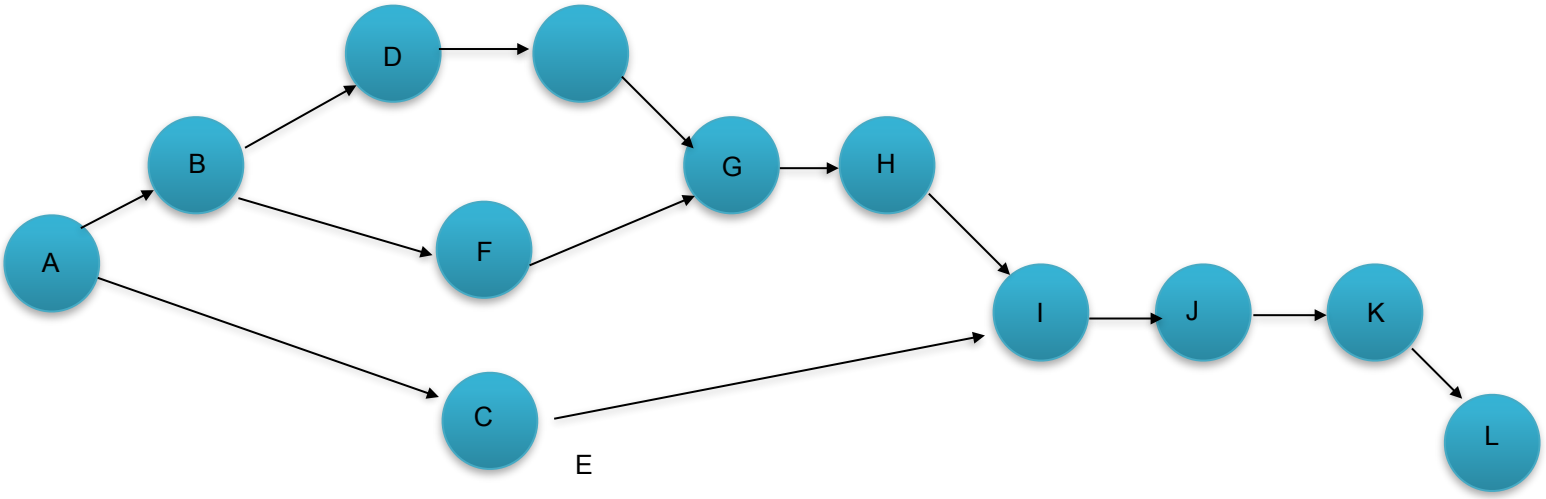
5.1. Work Breakdown Structure



5.2. Activity Network

Activity	Description	Immediate Predecessor	Estimated Time (days)
A	Plan project abstract and confirm objectives	–	2
B	Write down software requirements	A	5
C	Document the entire plan	A	30
D	Design and project planning	B	4
E	User interface design	D	15
F	Software development	B	15
G	Developed software and designed interface integration	E,F	10
H	Software implementation as a project	G	5
I	Development of software testing plan	C,H	10

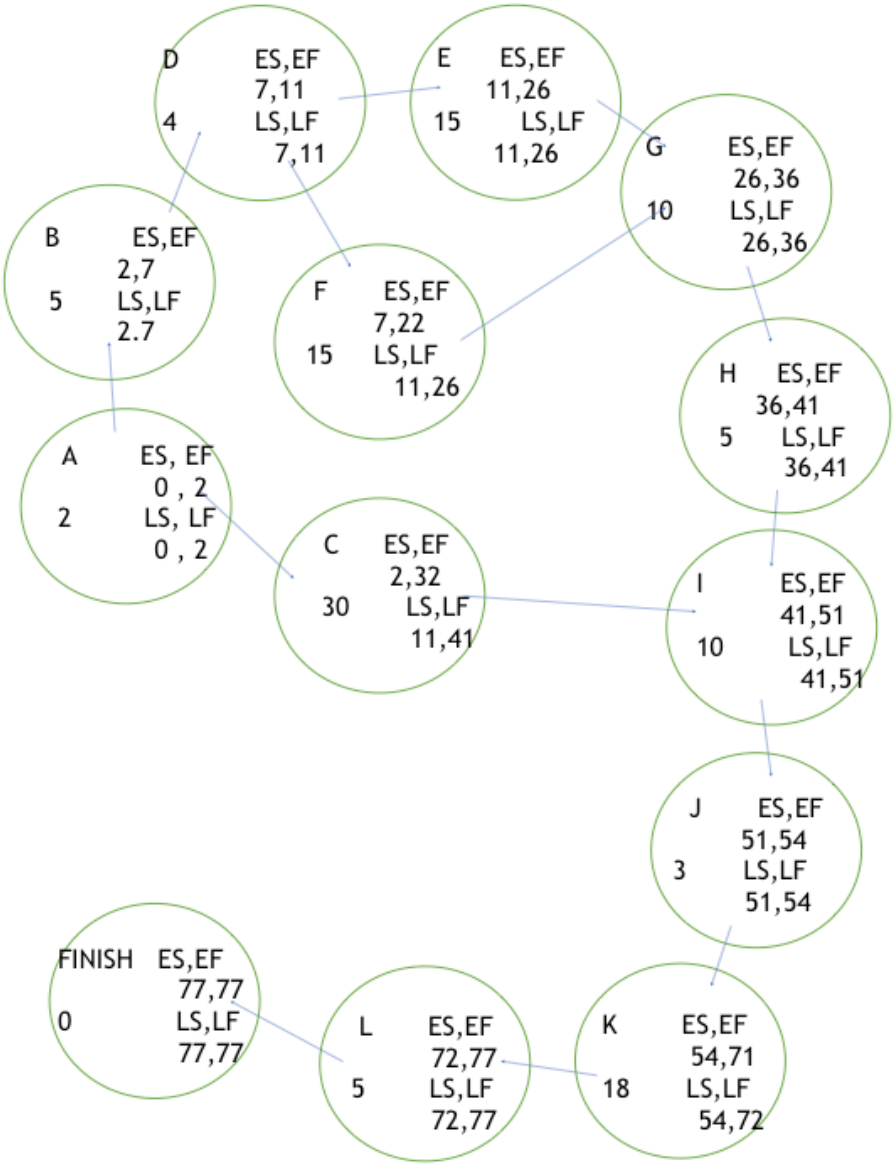
J	Test the first release	I	3
K	Software customization	J	18
L	Final testing	K	5



5.3. Critical Path Method

The expected project completion time is 77 days, given by the maximum EFT or LFT.

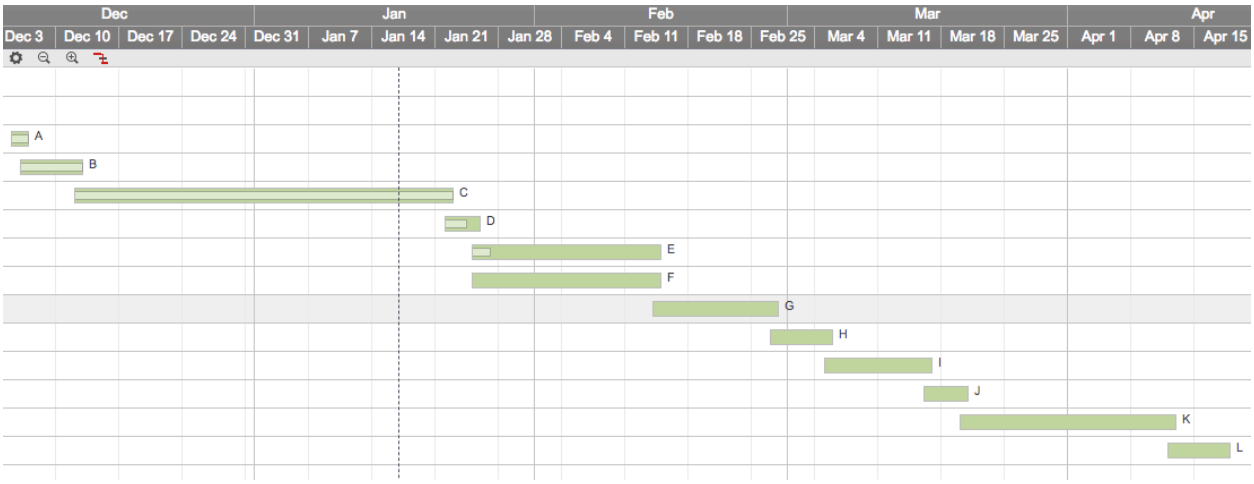
Critical Path: A-B-D-E-G-H-I-J-K-L-Finish



Activity	EST	EFT	LST	LFT	Stack	CA
A	0	2	0	2	0	Yes
B	2	7	2	7	0	Yes
C	2	32	11	41	9	
D	7	11	7	11	0	Yes
E	11	26	11	26	0	Yes
F	7	22	11	26	4	
G	26	36	26	36	0	Yes
H	36	41	36	41	0	Yes

I	41	51	41	51	0	Yes
J	51	54	51	54	0	Yes
K	54	72	54	72	0	Yes
L	72	77	72	77	0	Yes
Finish	77	77	77	77	0	Yes

5.4. Gantt Chart

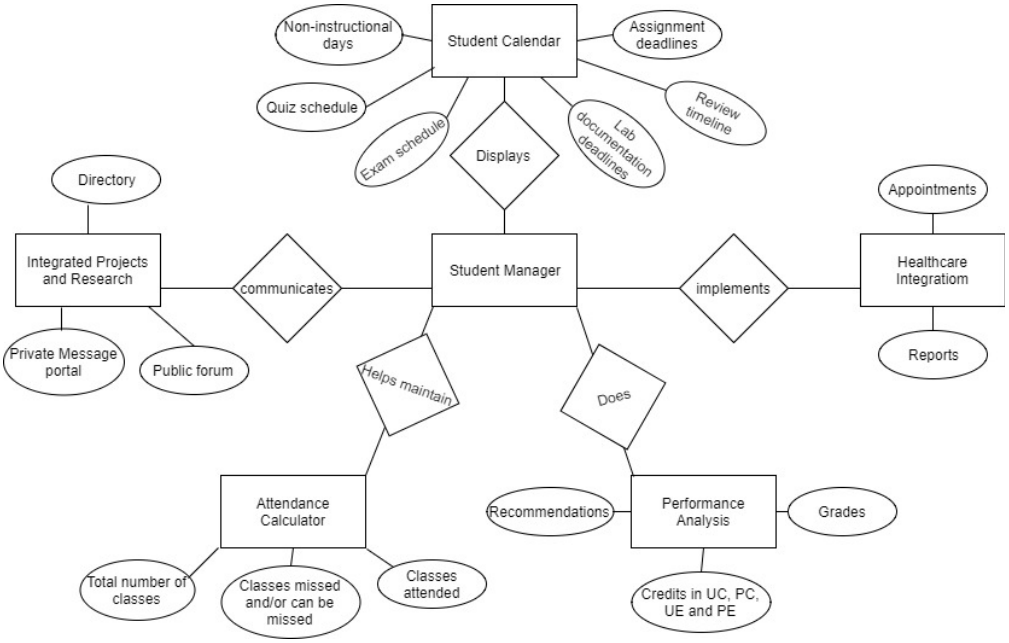


5.5. Process Model

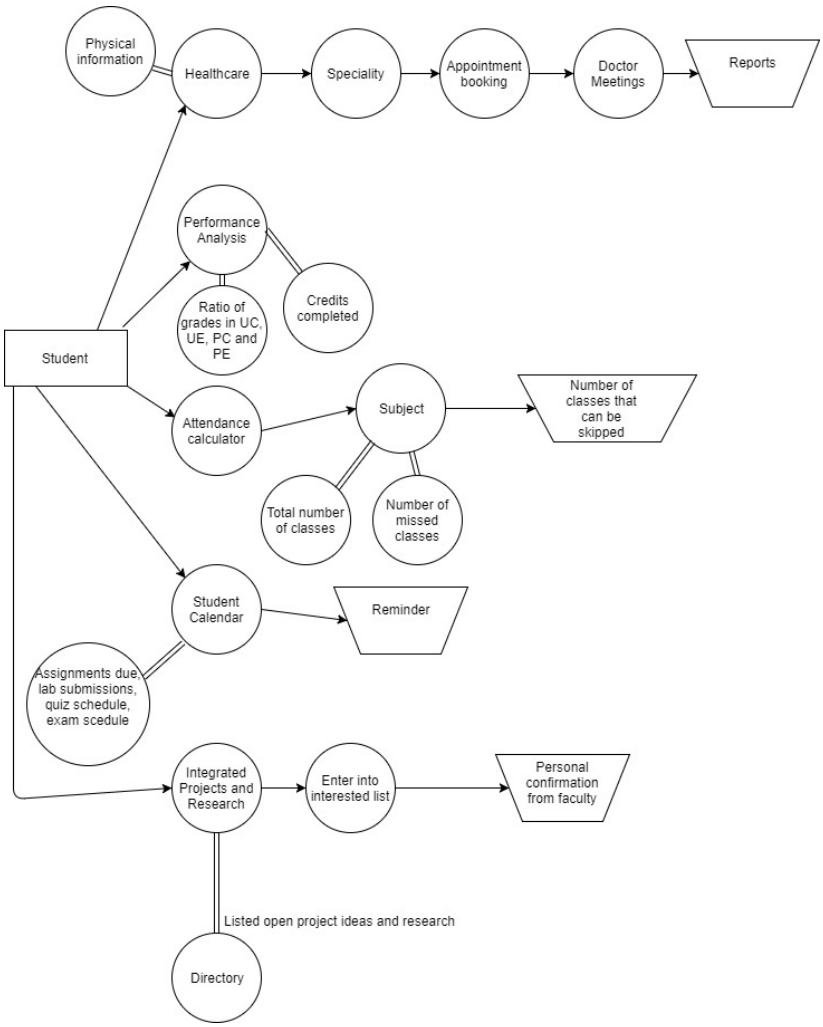
We are employing the iterative waterfall process model. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of software development model is basically used for the for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model software testing starts only after the development is complete. In waterfall model phases do not overlap. There is a feedback path provided to the past phases.

In our project, the requirements are clear, or follow a very structured process as in critical systems which needs a detailed, precise, and accurate documents describes the system to be produced. The requirements aren't ambiguous, and don’t support frequent interaction with the customers for feedback and proposing changes. It’s not a large project that might take long time to be developed and delivered either.

5.6. ER Diagram



5.7.DATA FLOW DIAGRAM



5.8.DATA DICTIONARY

health = p + n + a
p: string *input string*
n: string *input string*
a: string *input string*

performance = c + r + rs
c: integer *input number*
r: integer *input number*
rs: string

subjectattend = t + m+ more
t: integer *input number*
m: integer *input number*
more: integer

studcal = assign + lab + quiz + exam + rem
assign: integer *input number*
lab: integer *input number*
quiz: integer *input number*
exam: integer *input number*
rem: string

projres = dir + int
dir: string *input string*
int: string

err: string *error message*