Software Requirements Specification for : subtitle describing software

October 5, 2023

Contents

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 Purpose of the Project

1.1 User Business

The users would be students in high school to college institutes all over the world dealing with multiple courses.

1.2 Goals of the Project

We aim to develop a tool to help students integrate multiple course information with their personal schedules to generate a customized study plan. The plan should dynamically self-adjust according to the user's preference and study efficiency so that students can finish their deliverables before the deadline at their own pace with minimal pressure.

2 Stakeholders

2.1 Client

Insert your content here.

2.2 Customer

Insert your content here.

2.3 Other Stakeholders

Insert your content here.

2.4 Hands-On Users of the Project

Insert your content here.

2.5 Personas

Insert your content here.

2.6 Priorities Assigned to Users

Insert your content here.

2.7 User Participation

Insert your content here.

2.8 Maintenance Users and Service Technicians

Insert your content here.

3 Mandated Constraints

3.1 Solution Constraints

The development of a fully functional product is required to be finished by February 5 when the Revision 0 Demonstration is scheduled.

3.2 Implementation Environment of the Current System

The project implementation would be done through VS Code in the beginning and converted into Github Codespace once set up gets finished to ensure consistent compiling performance among group members.

3.3 Partner or Collaborative Applications

Our system would import event data from and export generated studen plan to popular calendar applications like Google Calendar and Outlook Calendar.

3.4 Off-the-Shelf Software

3.4.1 StudySchedule.org

StudySchedule is a free scheduling software dedicated to generating customized daily schedules for students to study for MCAT. They would ask the user to set up an account, pick study material from their MCAT resource

library, and take a questionnaire on time constraints and pace preference. Students could view their progress and make adjustments as they wish.

3.4.2 Taskade AI Genrator

Taskade is a powerful team project management tool. The component Taskade AI is capable of generating tasks for given project topics, creating checklists, project plans, and calendar schedules. Taskade AI is also capable of summarising PDF files.

3.5 Anticipated Workplace Environment

Our web-based app is anticipated to be compatible with mainstream browsers including Chrome, Safari, Microsoft Edge, and Firefox on the latest version of Windows and macOS.

3.6 Schedule Constraints

Each member of our team would devote 8 hours per week to work on the project making a total of 40 hours per week.

3.7 Budget Constraints

The monetary expenditure for the entire project could not exceed $750.

3.8 Enterprise Constraints

Insert your content here.

4 Naming Conventions and Terminology

4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

Insert your content here.

5 Relevant Facts And Assumptions

5.1 Relevant Facts

Manually reading through multiple course outlines, inputting all the deliverable information into the calendar, and crunch time has always been an inefficient part of academic life. Students wish to have a seamless tool integrating deadline management into their everyday lives without risking overdue penalties.

5.2 Business Rules

Insert your content here.

5.3 Assumptions

5.3.1

Course outlines are available as PDF files.

5.3.2

Users are using a mainstream Calendar application: Google Calendar, Outlook Calendar, Calendar.

5.3.3

Users have a relatively stable weekly schedule.

6 The Scope of the Work

6.1 The Current Situation

Insert your content here.

6.2 The Context of the Work

Insert your content here.

6.3 Work Partitioning

Insert your content here.

6.4 Specifying a Business Use Case (BUC)

Insert your content here.

7 Business Data Model and Data Dictionary

7.1 Business Data Model

Insert your content here.

7.2 Data Dictionary

7.2.1 Users

Users		
Attribute	userName	userPwd
Description	Unique identifier of a user	The password of an account
Type	`VARCHAR(50)`	`VARCHAR(50)`
Allowed Values		
Default Value		
Constraints	`PRIMARY KEY, NOT NULL`	`NOT NULL CHECK (CHAR_LENGTH(userPwd) >8)`
Source	User input when setting up the account	User input when setting up the account
Usage	Authentication	Authentication

7.2.2 Courses

Courses		
Attribute	subject	courseCode
Description	The subject of a course	The code of a course
Type	`VARCHAR(16)`	`VARCHAR(16)`
Allowed Values		
Default Value		
Constraints	`NOT NULL, PART OF PRIMARY KEY`	`NOT NULL, PART OF PRIMARY KEY`
Source	Extracted from course outline	Extracted from the course outline
Usage	Record course information	Record course information
Attribute	courseId	
Description	The unique identifier of a course	
Type	`VARCHAR(16)`	
Allowed Values		
Default Value		
Constraints	`PRIMARY KEY`	
Source	Uniquely generated when a course outline is uploaded	
Usage	Record course information	

7.2.3 Tasks

Tasks		
Attribute	weight	taskType
Description	The percentage weight associated with a task	The type of a task
Type	`DECIMAL(10,2)`	`INT`

Allowed Values		`0 - QUIZ`
		`1 - ASSIGNMENT`
		`2 - PRESENTATION`
		`3 - MIDTERM`
		`4 - EXAM`
		`5 - REPORT`
		`6 - OTHER`

Default Value	0	0
Constraints	`NOT NULL CHECK (weight >= 0 AND weight <= 100)`	`NOT NULL`
Source	Extracted from the course outline	Extracted from the course outline
Usage	Record course information	Record course information
Attribute	subject	courseCode
Description	The subject of a course	The code of a course
Type	`VARCHAR(16)`	`VARCHAR(16)`
Allowed Values		
Default Value		
Constraints	`NOT NULL`	`NOT NULL`
Source	Extracted from course outline	Extracted from the course outline
Usage	Record course information	Record course information
Attribute	courseId	
Description	The unique identifier of a course	
Type	`VARCHAR(16)`	
Allowed Values		
Default Value		
Constraints		
Source	Uniquely generated when a course outline is uploaded	
Usage	Record course information	

7.2.4 date

date	
Alias	
Description	A date
Type	`DATE`
Allowed Values	
Default Value	
Constraints	`NOT NULL`
Source	Extracted from the course outline
Usage	Record course information

8 The Scope of the Product

8.1 Product Boundary

Insert your content here.

8.2 Product Use Case Table

Insert your content here.

8.3 Individual Product Use Cases (PUC's)

Insert your content here.

9 Functional Requirements

9.1 Authentication

9.1.1

The user could create an account with a username and a password.

9.1.2

The user could log in to their account by providing a corresponding username and password.

9.1.3

The user could access their and only their scheduling information once logged in.

9.2 User input

9.2.1

Without overwriting, the user could upload multiple PDF files containing course outlines.

9.2.2

The system prompts users to choose their preferred study interval at which the system tends to allocate study sessions and allows changes later.

9.2.3

The system prompts users to set their preferred Pomodoro intervals and allows changes later.

9.2.4

The system supports a friend list allowing users to send/reject/accept friend requests.

9.2.5

On finishing each sub-task, the system would ask for the user's feedback on whether the pace i comfortable.

9.3 Data

9.3.1

The system could extract information including `courseName, taskType, taskName, weight, deadline` from uploaded course outline *.pdf* files.

9.3.2

With authentication, the system could import event and schedule data from other calendar apps including Calendar, Outlook, and Google Calendar.

9.3.3

With authentication, the system could export event and schedule data to other calendar apps including Calendar, Outlook, and Google Calendar.

9.4 Scheduling

9.4.1

The system could calculate the estimated time needed to finish each task extracted from course outlines.

9.4.2

The estimated time needed to finish each task would dynamically adjust based on user feedback on past task completion progress.

9.4.3

The system could prioritize tasks extracted based on `taskType, weight` and `deadline`.

9.4.4

With data containing task information, preferred study time, schedule, and Pomodoro intervals gathered the system could generate a detailed study plan containing multiple sub-tasks for each task in the course outline available in-app and export it to other calendars.

9.4.5

The system would have a Pomodoro clock ready for each sub-task.

9.4.6

The system allows users to check their progress on each task and adjust the time needed.

9.4.7

With a detailed study plan, the system could pair friends from the friend list with similar study plans to have video-based online study sessions.

10 Look and Feel Requirements

10.1 Appearance Requirements

Insert your content here.

10.2 Style Requirements

Insert your content here.

11 Usability and Humanity Requirements

11.1 Ease of Use Requirements

The system must be easy to use by users with at least a grade 9 education background.

11.2 Personalization and Internationalization Requirements

N/A

11.3 Learning Requirements

The system must be understood by users within 10 minutes of exploring.

11.4 Understandability and Politeness Requirements

11.4.1

The language in the app must be grammatically correct 99% of the time.

11.4.2

The language in the app must be non-offensive.

11.5 Accessibility Requirements

Color combinations used in the interface must be distinguished by users with color blindness.

12 Performance Requirements

12.1 Speed and Latency Requirements

Insert your content here.

12.2 Safety-Critical Requirements

Insert your content here.

12.3 Precision or Accuracy Requirements

Insert your content here.

12.4 Robustness or Fault-Tolerance Requirements

Insert your content here.

12.5 Capacity Requirements

Insert your content here.

12.6 Scalability or Extensibility Requirements

Insert your content here.

12.7 Longevity Requirements

Insert your content here.

13 Operational and Environmental Requirements

13.1 Expected Physical Environment

The system must be operable under the same physical environment that the desktop computer running it is operable.

13.2 Requirements for Interfacing with Adjacent Systems

The system could interface with calendar APIs when called.

13.3 Productization Requirements

N/A

13.4 Release Requirements

The released version must pass all known regression tests.

14 Maintainability and Support Requirements

14.1 Maintenance Requirements

Insert your content here.

14.2 Supportability Requirements

Insert your content here.

14.3 Adaptability Requirements

Insert your content here.

15 Security Requirements

15.1 Access Requirements

15.1.1

The system is accessible only if the correct combo of username and password is provided.

15.1.2

Users could not access other users' data.

15.2 Integrity Requirements

No Unauthorised entity could modify the database.

15.3 Privacy Requirements

15.3.1

The system will not release user information to a third party.

15.3.2

user authentication information will be stored with encryption.

15.4 Audit Requirements

N/A

15.5 Immunity Requirements

N/A

16 Cultural Requirements

16.1 Cultural Requirements

Insert your content here.

17 Compliance Requirements

17.1 Legal Requirements

The system must comply with local laws and regulations.

17.2 Standards Compliance Requirements

The code must comply with *Flack8* coding standards.

18 Open Issues

Insert your content here.

19 Off-the-Shelf Solutions

19.1 Ready-Made Products

Insert your content here.

19.2 Reusable Components

Insert your content here.

19.3 Products That Can Be Copied

Insert your content here.

20 New Problems

20.1 Effects on the Current Environment

Insert your content here.

20.2 Effects on the Installed Systems

Insert your content here.

20.3 Potential User Problems

Insert your content here.

20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Insert your content here.

20.5 Follow-Up Problems

Insert your content here.

21 Tasks

21.1 Project Planning

Insert your content here.

21.2 Planning of the Development Phases

Insert your content here.

22 Migration to the New Product

22.1 Requirements for Migration to the New Product

Insert your content here.

22.2 Data That Has to be Modified or Translated for the New System

Insert your content here.

23 Costs

Insert your content here.

24 User Documentation and Training

24.1 User Documentation Requirements

Insert your content here.

24.2 Training Requirements

Insert your content here.

25 Waiting Room

Insert your content here.

26 Ideas for Solution

Insert your content here.

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?