

1. Cover page & 2. Inside Front page (see sample given)
SCRUM PROJECT DOCUMENTATION TITLES (2024)
TITLE(16 Bold all capitals – Times New Roman)

PROJECT REPORT
(12 Regular- all capitals)

submitted by
NAME: (14 Bold- all capitals)
SARAH P
Reg. No (23MCA1101 12 Bold)
PRIYA B
Reg. No (23MCA1110 12 Bold)
DAVID S (23MCA1101)
Reg. No (23MCA1141 12 Bold)

to
the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree
of
Master of Technology in (specialization) (12 Italics)



Department of Computer Applications
(14 Bold, leading capitals)
Sree Narayana Gurukulam College of Engineering (SNGCE)
Name of college
Kadayiruppu
November, 2024
(14 Regular)

3. **DECLARATION** (see sample given)

4. **CERTIFICATE** (see sample given)

5. CONTENTS (Main headings, Sub headings & page no. shall be given) (see sample given)

6. ACKNOWLEDGEMENT

7. ABSTRACT (Do not exceed 2 pages – use double or 1.5 line spacing with 10 or 12 Regular font) (No figures, sketches, tables shall be there)

8. LIST OF TABLES (No. , Title and Page No shall be given)

9 . LIST OF FIGURES (No. , Title and Page No shall be given)

10. ABBREVIATIONS: List all abbreviations and their expansions

11. NOTATION: List all symbols used in the report, give units in case of dimensional quantities

12. SCRUM PROJECT Details and relevance of scrum used in academics

Curriculum Objectives and its execution and mappings.

13. TEAM ROLES AND RESPONSIBILITIES

Team structure identified and detailed.

14 CHAPTERS

APPENDICES (if any)

REFERENCES

Note:-

Chapter Titles- 16 Bold – All capitals – No underlining

Main headings - 14 Bold - All capitals – No underlining

Main headings shall be numbered as 1.1,1.2,1.3,..... 2.1,2.2,2.3..... etc.

Sub headings- 12 Bold- Leading capitals- No underlining

Sub headings shall be numbered as 1.1.1,1.1.2,.....2.1.1,2.1.2.....etc

Sub-sub headings – 12 Regular- Leading capitals –shall be numbered as. (i), (ii) ...

Figures, sketches, equations and tables shall be serially numbered chapter wise

(Eg. 2.1,2.2..... 3.1,3.2.....).

- All figures, sketches, photos and tables shall be titled.

- Figure no. and title (12 Regular) shall be given below the figure.
- Table no and title (12 Regular) shall be given above the table.
- Figures, Tables etc in Landscape format shall be put in such a way that they can be viewed from right side
- Give reference no within square brackets for figures, sketches, photos tables which are adapted from the references.
- Text – 12 Regular, Times New Roman, double spacing, Alignment- Justified.
- Headers or footers not required

Figures, Tables, Sketches and Equations shall be **centre justified**. Figures, sketches, tables shall be placed immediately after the paragraph in which they are referred.

Use equation editor for equations.

Begin each chapter in a new page.

Begin paragraph in the line next to the heading. Leave one line space after each paragraph. Leave one line space after paragraph and the next heading. Do not begin a new section at the end of a page. Minimum two lines must follow a main heading/sub heading in a page.

Begin all paragraphs left justified. Leave 12 pt space after paragraphs. OR. Start paragraphs indenting 10 character space in the beginning and in this case no space is to be given after paragraph.

References shall be as per the following format

- **Journal/conference/symposium/seminar/workshop papers:-** Authors (in bold) (Year), Title of paper, Name of Journal/ Conference/ Symposium/ seminar/ workshop (in italics), Issue No., Pages
- **Books :** -Author(s) (Bold), Title (in italics), Publisher, Edition, Year of Pubn. Online Books: - Author.(year, month day).
- Title. (edition) [Type of medium].
Volume (issue). Available: site/path/file
- Patents: - Author, Title of patent, Patent No., Month day, year

- **Page numbering : Arabic numerals (12 Regular font) – bottom centred.**

Start page number 1 from Chapter 1. Page numbers shall not be shown on Chapter beginning pages.

- **Important**

(i) A typed draft report as per the above guidelines has to be prepared and submitted to the guide(s), at least one week before the final evaluation of the project. The draft report shall be corrected and approved by the guide(s). This, signed draft report is to be produced before the evaluation committee at the time of final evaluation of the project.

(ii) The final report is to be made after the final project evaluation is over. The corrections and suggestions made by the evaluation committee are to be incorporated in the final report. Submit the final report along with the draft report ,within one week after final project evaluation, to the Project coordinator for getting signature of the Head of the Dept.

Note: - A sample report is given. This report is only a sample and the titles and contents are randomly included and have no significance other than conveying the format.

SCRUM PROJECT: DEVELOPMENT OF A TASK MANAGEMENT APPLICATION

PROJECT REPORT

submitted by

SARAH P
(23MCA1101)
PRIYA B
(23MCA1110)
DAVID S
(23MCA1141)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications



Department of Computer Applications

Sree Narayana Gurukulam College of Engineering (SNGCE)
Kadayiruppu

November, 2024

DECLARATION

I undersigned hereby declare that the project report (“**Development of a Task Management Application**”), submitted for partial fulfillment of the requirements for the award of degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of (Name of supervisor(s)). This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University. (12 Regular, 1.5 line spacing).

Place
Date

Signature
Name of the student

DEPARTMENT OF COMPUTER APPLICATIONS
SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING
(SNGCE)
KADAYIRUPPU.
(14-Bold.)



CERTIFICATE (14 Bold)

This is to certify that the report entitled “**Title**” (14 Bold)” submitted by “**Name**” (12 Bold) to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Technology in (stream & branch) is a bonafide record of the project work carried out by him/her under my/our guidance and supervision.. This report in any form has not been submitted to any other University or Institute for any purpose. (12 Regular, 1.5 line spacing).

Internal Supervisor(s)

External Supervisor(s)
(if any)

Project Coordinator

HEAD OF THE DEPT

Contents		
No	Title	Page
	ACKNOWLEDGEMENT	i
	ABSTRACT	ii
	LIST OF TABLES	iv
	LIST OF FIGURES	vi
	ABBREVIATIONS	vii
	NOTATION	ix
	SCRUM PROJECT	x
	TEAM ROLES AND RESPONSIBILITIES	xii
Chapter 1.	INTRODUCTION	1
	1.1 Project Overview	
Chapter 2.	REQUIREMENTS/ SURVEYS	3
	2.1 Problem Scenario	
	2.2 Hardware Requirements	
	2.3 Software Requirements	
	2.4 Functional Requirements	
	2.5 Non-Functional Requirements	
	2.6 Product Functions	
Chapter 3.	ANALYSIS	10
	3.1 Feasibility Analysis	
	3.2 Problem Story	
	3.3 Use Case Story	
	3.4 Use Case Diagrams	
	3.5 User Stories	
	3.6 Technical Documentation	
Chapter 4.	PLANNING	17
	4.1 Product Backlog	
	4.2 Sprint Planning Documents	
	4.3 Sprint Backlog	
	4.4 Project Plans	
	4.5 Ideal Burn Down Chart	
Chapter 5.	DESIGNS	25
	5.1 UID Design Documentation	
	5.2 Data Design Documentation	
	5.3 Flow Designs, Activity Designs, Sequence Designs etc...	
Chapter 6.	CODING/ DEVELOPMENT	35
	6.1 Increment Definition	
	6.2 Code Samples	
Chapter 7.	TESTING	45
	7.1 Testing Documentation	
	7.2 Test plans derived	

	7.3 Results from testing	
Chapter 8. DEPLOYMENT STRATEGY		55
	8.1Version Control and Git	
	8.2 Release Notes	
Chapter 9. SUMMARY		52
	9.1 Retrospective Notes	
APPENDICES		58
REFERENCES		

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to all those who have contributed to the successful completion of our academic mini project, "Development of a Task Management Application," undertaken as part of our curriculum at SNGCE affiliated with Kerala Technological University (KTU).

First and foremost, we wish to extend our sincere thanks to **[Principal's Name]**, Principal of SNGCE, for fostering an environment that encourages innovation and learning. Their commitment to academic excellence has provided us with the resources and opportunities necessary for our growth.

We are also deeply grateful to, **Dr. [Head of the Department's Name]**, Head of the Department of [Department Name], for their unwavering support and guidance throughout this project. Their leadership and vision have greatly inspired us to strive for excellence.

Our special thanks go to our project guide, **Dr. /Ms [Guide's Name, designation]**, whose expertise and mentorship were invaluable throughout the development process. Their insights and constructive feedback have helped us refine our approach and achieve our project objectives.

We would like to acknowledge the contributions of other faculty members, including **[Other Teachers' Names]**, who provided us with essential knowledge and support during various stages of our project. Their dedication to teaching has significantly enriched our learning experience.

We are profoundly thankful to **God Almighty** for granting us the strength, wisdom, and perseverance needed to complete this project. We believe that faith has played a crucial role in guiding us through challenges. Lastly, we wish to express our heartfelt appreciation to our parents for their unconditional love, encouragement, and support throughout our academic journey. Their belief in us has been a constant source of motivation.

This project has not only enhanced our technical skills but also taught us valuable lessons in teamwork and project management. We are excited to carry these experiences forward into our future endeavours.

ABSTRACT

The **Development of a Task Management Application** project aims to address the growing need for efficient task organization and productivity enhancement in both personal and professional environments. As remote work and digital collaboration become more prevalent, individuals and teams face challenges in managing tasks effectively, often leading to missed deadlines and decreased productivity. This application is designed to provide users with an intuitive interface that facilitates the creation, tracking, and management of tasks while incorporating essential features such as categorization, deadlines, and notifications.

Utilizing the Scrum framework, the project emphasizes iterative development and continuous feedback, ensuring that user needs are met throughout the development process. The application will be built using modern technologies, including React.js for the front end, Node.js for the back end, and MongoDB for data storage. Key functionalities will encompass user authentication, task prioritization, and reminder systems, all aimed at enhancing user experience.

The project's objectives include delivering a scalable solution that not only simplifies task management but also integrates seamlessly into users' workflows. By conducting thorough analysis and user research, the application will be tailored to meet the specific requirements of diverse stakeholders. This project ultimately seeks to empower users to manage their tasks more effectively, thereby improving overall productivity and satisfaction.

LIST OF TABLES

No.	Title	Page No.
2.1	XXXXXXXXXXXXXXXXXXXXX	9
2.2	XXXXXXXXXXXXXXXXXXXXX	15
.....		
3.1	XXXXXXXXXXXXXXXXXXXXX	
3.2	XXXXXXXXXXXXXXXXXXXXX	
3.3	XXXXXXXXXXXXXXXXXXXXX	
.....		
4.1	XXXXXXXXXXXXXXXXXXXXX	
.....		
.....		

LIST OF FIGURES

No.	Title	Page No.
2.1	XXXXXXXXXXXXXXXXXXXXX	8
2.2	XXXXXXXXXXXXXXXXXXXXX	12
.....		
3.1	XXXXXXXXXXXXXXXXXXXXX	
3.2	XXXXXXXXXXXXXXXXXXXXX	
3.3	XXXXXXXXXXXXXXXXXXXXX	
.....		
4.1	XXXXXXXXXXXXXXXXXXXXX	
.....		

.....

ABBREVIATIONS

(List in the alphabetical order)

HAS High Altitude Simulation

LMTD Logarithmic Mean Temperature Difference

PDF Probability Density Function

.....

.....

NOTATION

(List in the alphabetical order)

A	Area, m^2
E	Voltage, V
Re	Reynolds number
T	Temperature, K

.....

Greek Symbols

α	Diffusivity, m^2/s
τ	Shear stress, MPa

SCRUM PROJECT

INTRODUCTION TO SCRUM

Scrum is an Agile framework designed for managing complex projects, particularly in software development. It emphasizes iterative progress through a series of time-boxed iterations known as sprints. Scrum promotes collaboration, accountability, and continuous improvement.

OVERVIEW OF ACADEMIC MINI PROJECTS USING SCRUM AT APJA-KTU

Project Structure: Students typically form small teams to work on a mini project that addresses a specific problem or need within their domain of study. The projects can range from software development to research initiatives, allowing students to apply theoretical knowledge in practical scenarios.

Roles and Responsibilities: Each team member assumes specific roles within the Scrum framework:

- **Product Owner:** If there is a customer for the project then he/she will be the Product Owner (External Guide/ HoD) and a faculty from the department will be the Internal Guide. If there is no such customer then the Internal Guide himself/herself shall act as the Product Owner. This role is usually filled by those who represents the stakeholders' interests, prioritizes the project backlog, and defines the project vision.
- **Scrum Master:** A faculty / technical staff shall act as the Scrum Master to continuously monitor the project development. Periodic meetings, of less than 15 minutes, at the convenience of the Scrum Master are to be highly encouraged. SM facilitates Scrum practices, helps resolve impediments, and ensures the team adheres to Agile principles.
- **Development Team:** Comprised of students with diverse skill sets, this group collaborates to deliver project increments during each sprint.

Execution Process

The project is executed in iterative cycles called sprints, typically lasting 2-4 weeks. Each sprint begins with planning sessions where user stories from the backlog are selected for

development. Daily stand-up meetings foster communication and allow team members to discuss progress and challenges. At the end of each sprint, a review meeting is held to demonstrate completed work and gather feedback from stakeholders.

LEARNING OUTCOMES

By using Scrum for their mini projects, students at KTU gain hands-on experience in Agile methodologies, enhance their teamwork skills, and improve their ability to manage time and resources effectively. This practical application of Scrum not only prepares them for future careers but also reinforces their understanding of project management principles.

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Identify a real-life project that is useful to society/industry	Level 2: Understand
CO 2	Interact with people to identify the project requirements	Level 3: Apply
CO 3	Apply suitable development methodology for the development of the product/project	Level 3: Apply
CO 4	Analyse and design a software product/project	Level 4: Analyse
CO 5	Test the modules at various stages of project development	Level 5: Evaluate
CO 6	Build and integrate different software modules	Level 6: Create
CO 7	Document and deploy the product/project	Level 3: Apply

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	2	3	3	3	1	2	3	3	3	3	3	3
CO 2	2	3	2	3	2	3	2	1	3	2	3	
CO 3	3	3	3	3	3	1	3	3	1	2		
CO 4	3	3	3	3	3	3	3	1	1	2		
CO 5	3	3	3	3	3	2	3	1				
CO 6	3	3	3	3	3	2	3	3	2	3	3	
CO 7	1	1	3	3	3	2	3	3	2	1	2	

3/2/1: High/Medium/Low

TEAM ROLES AND RESPONSIBILITIES

In an academic project utilizing the Scrum framework, the roles and responsibilities of the team members are crucial for ensuring effective collaboration and successful project outcomes. The Scrum team is composed of three primary roles: **Product Owner**, **Scrum Master**, and **Development Team**.

Product Owner

The Product Owner acts as the voice of the stakeholders and is responsible for defining the vision of the project. This role involves managing and prioritizing the product backlog, which is a dynamic list of features, enhancements, and fixes that need to be addressed. The Product Owner ensures that the team focuses on delivering maximum value by clearly articulating user stories and acceptance criteria, thus guiding the Development Team on what needs to be built next.

Scrum Master

The Scrum Master serves as a facilitator and coach for the Scrum team. Their primary responsibilities include ensuring that Scrum practices are followed, helping to remove any impediments that may hinder the team's progress, and fostering an environment conducive to collaboration and continuous improvement. The Scrum Master also protects the team from external distractions and encourages open communication among team members, ensuring that everyone understands their roles within the Scrum process.

Development Team

The Development Team consists of cross-functional members who are responsible for delivering potentially shippable increments of the product at the end of each sprint. This team is self-organizing, meaning they have the autonomy to decide how best to accomplish their work. Members typically include software developers, designers, testers, and other specialists who collaborate closely to meet sprint goals. The Development Team is accountable for maintaining high-quality standards and adhering to the "Definition of Done" for each increment.

List of Team Members

- **Product Owner:** _____
- **Scrum Master:** _____
- **Development Team Members:**

- [1]. **SARAH P** (Responsibilities held in project)
- [2]. _____ (Responsibilities held in project)
- [3]. _____ (Responsibilities held in project)

CHAPTER 1. INTRODUCTION

1.1 PROJECT OVERVIEW

CHAPTER 2. REQUIREMENTS/ SURVEYS

2.1 PROBLEM SCENARIO

2.2 HARDWARE REQUIREMENTS

2.3 SOFTWARE REQUIREMENTS

2.4 FUNCTIONAL REQUIREMENTS

2.5 NON-FUNCTIONAL REQUIREMENTS

2.6 PRODUCT FUNCTIONS

CHAPTER 3. ANALYSIS

3.1 FEASIBILITY ANALYSIS

3.2 PROBLEM STORY

3.3 USE CASE STORY

3.4 USE CASE DIAGRAMS

3.5 USER STORIES

3.6 TECHNICAL DOCUMENTATION

CHAPTER 4. PLANNING

4.1 PRODUCT BACKLOG

4.2 SPRINT PLANNING DOCUMENTS

4.3 SPRINT BACKLOG

4.4 PROJECT PLANS

4.5 IDEAL BURNDOWN CHART

CHAPTER 5. DESIGNS

5.1 UID DESIGN DOCUMENTATION

5.2 DATA DESIGN DOCUMENTATION

5.3 FLOW DESIGNS, ACTIVITY DESIGNS, SEQUENCE DESIGNS ETC...

CHAPTER 6. CODING/ DEVELOPMENT

6.1 INCREMENT DEFINITION

6.2 CODE SAMPLES

CHAPTER 7. TESTING

7.1 TESTING DOCUMENTATION

7.2 TEST PLANS DERIVED

7.3 RESULTS FROM TESTING

CHAPTER 8. DEPLOYMENT STRATEGY

8.1 VERSION CONTROL AND GIT

8.2 RELEASE NOTES

CHAPTER 9. SUMMARY

9.1 RETROSPECTIVE NOTES

REFERENCES

APPENDICES

CHAPTER 1. INTRODUCTION

1.1 PROJECT OVERVIEW

Brief Description of the Project

The project aims to develop a **Task Management Application** that allows users to create, track, and manage tasks efficiently. The application will feature user authentication, task categorization, deadlines, and reminders. **Objectives and Goals**

- To provide a user-friendly interface for task management.
- To implement features that enhance productivity, such as notifications and progress tracking.
- To ensure the application is scalable and maintainable.

CHAPTER 2. REQUIREMENTS/ SURVEYS

2.1 PROBLEM SCENARIO

Description of the Issue or Challenge the Project Addresses

Many individuals and teams struggle with managing tasks effectively, leading to missed deadlines and decreased productivity. Existing solutions often lack essential features or are overly complex.

Context and Background Information

With the rise of remote work and digital collaboration, there is an increasing demand for tools that facilitate task management across various platforms.

Table 2.3 Effect of voltage
(Title of table is to be given above the table)

Sl. No	Voltage (mV)	Current (mA)	Force (N)	Power (W)
1	nn	mm	xx.x	yy.yy
2	nn	mm	xx.x	yy.yy
3	nn	mm	xx.x	yy.yy
4	nn	mm	xx.x	yy.yy
5	nn	mm	xx.x	yy.yy
6	nn	mm	xx.x	yy.yy
7	nn	mm	xx.x	yy.yy

.....
.....

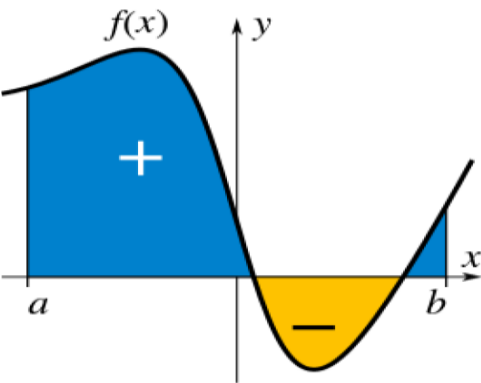


Fig.2.1 Variation of xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxx (David, 2014)
(Title of figures to be given below the figure)

2.2 HARDWARE REQUIREMENTS

2.3 SOFTWARE REQUIREMENTS

2.4 FUNCTIONAL REQUIREMENTS

2.5 NON-FUNCTIONAL REQUIREMENTS

2.6 PRODUCT FUNCTIONS

The **Task Management Application** developed as part of our academic mini project at Kerala Technological University (KTU) serves several essential functions aimed at enhancing productivity and organization for users. Utilizing the Scrum framework, the application is designed to address common challenges faced in task management through a user-friendly interface and robust features.

2.6.1 CORE FUNCTIONS

1. Task Creation and Organization

Users can easily create tasks by entering relevant details such as titles, descriptions, due dates, and priorities. This feature allows for quick capture and organization of tasks, ensuring that no important work is overlooked.

2. Task Assignment and Collaboration

The application facilitates team collaboration by enabling users to assign tasks to themselves or other team members. This functionality includes features for commenting, attaching files, and discussing tasks directly within the application, promoting seamless communication among team members.

3. Task Prioritization and Deadlines

Users can prioritize tasks based on urgency or importance, helping them focus on critical work. Setting deadlines ensures that tasks are completed on time, which is vital for maintaining project schedules.

4. Progress Tracking and Visualization

The application provides tools for tracking the progress of tasks through status updates (e.g., not started, in progress, completed) and visual progress indicators. This feature offers users a clear overview of project status, enabling timely interventions when necessary.

5. Automated Reminders and Notifications

Users receive automated reminders for upcoming deadlines and task updates, reducing the chances of missed deadlines and ensuring that everyone stays informed about task statuses.

6. Integration with Other Tools

The application supports integration with various productivity tools (such as calendars or communication platforms), streamlining workflows and enhancing overall efficiency.

7. Reporting and Analytics

Users can generate reports to analyse task completion rates, individual productivity, and overall project performance. This data-driven approach aids in identifying bottlenecks and areas for improvement.

CHAPTER 3. ANALYSIS

Examination of the Problem Scenario The current market offers numerous task management tools; however, they often fail to meet user needs regarding simplicity and functionality.

Identification of Key Stakeholders and Their Needs

- **End Users:** Require an intuitive interface and essential features for task tracking.
- **Project Managers:** Need tools for monitoring team productivity and task completion.

3.1 FEASIBILITY ANALYSIS

Feasibility Studies and Documentation: Initial feasibility studies indicate a strong market demand for a new solution that combines simplicity with powerful features.

1. Technical
2. Operational
3. etc...

3.2 PROBLEM STORY

Narrative Detailing the Problem from the User's Perspective "As a busy professional, I often forget important tasks because my current tool is too complicated to navigate quickly."

3.3. USE CASE STORY

Specific Scenarios Illustrating How Users Will Interact with the System Users will log in, create tasks with deadlines, categorize them by priority, and receive reminders as deadlines approach.

3.4 USE CASE DIAGRAMS

Visual representations will include actors such as **Users**, **Admin**, and their interactions with the system (task creation, modification).

3.5 USER STORIES

High-level requirements framed as user stories:

- "As a user, I want to create tasks so that I can keep track of my work."
- Acceptance criteria: Task creation should be simple with options for deadlines and categories.

3.6. TECHNICAL DOCUMENTATION

Technical Specifications or Architecture Documents Relevant to the Project The application will be built using a **React.js front-end**, a **Node.js back-end**, and a **MongoDB database**. **Guidelines for Maintaining Technical Quality Throughout Development**

- Implement code reviews for all pull requests.
- Use automated testing frameworks to ensure code quality.

Guidelines for Implementation and Integration Follow CI/CD practices to streamline deployment processes. (optional)

CHAPTER 4. PLANNING

Introduction to methods of planning used in the project

4.1 PRODUCT BACKLOG

A prioritized list will include user stories such as:

1. User authentication
2. Task creation
3. Task categorization
4. Notifications

4.2. SPRINT PLANNING DOCUMENTS

Overview of sprint goals:

- Sprint Goal: Implement user authentication.
- Selected backlog items: User login functionality, password recovery.

4.3. SPRINT BACKLOG

Detailed tasks:

- Set up user authentication (Assigned to Michael).
- Create login UI (Assigned to Sarah).

4.4 PROJECT PLANS

Overall project timeline includes milestones for each sprint cycle:

- Sprint 1: User Authentication
- Sprint 2: Task Creation Features
- Resource allocation involves assigning team members based on expertise.

Charts and Tables are to be included case to case.

4.5. IDEAL BURNDOWN CHART

Visual representation will show work completed versus remaining work over time to track sprint progress effectively.

CHAPTER 5. DESIGNS

Introduction to methods of designs used in the project

5.1 UID DESIGN DOCUMENTATION

Wireframes or Mockups of User Interfaces

Wireframes will illustrate key screens like the dashboard, task creation form, and settings page.

5.2 DATA DESIGN DOCUMENTATION

Data Models Including Entity-Relationship Diagrams (ERDs)

The ERD will include entities such as Users, Tasks, Categories, and Reminders with their relationships defined.

5.3 FLOW DESIGNS, ACTIVITY DESIGNS, SEQUENCE DESIGNS ETC...

CHAPTER 6. CODING/ DEVELOPMENT

Introduction to methods of coding and conventions used in the project

6.1 INCREMENT DEFINITION

Criteria for what constitutes a "done" increment:

- All acceptance criteria met.
- Code reviewed and tested successfully.

Milestones/ Configuration Items delivered/ Incremental work product details are to be included.

6.2 CODE SAMPLES

CHAPTER 7. TESTING

7.1 TESTING DOCUMENTATION

Test plans derived from acceptance criteria will include unit tests for each feature developed during sprints. **List down the tests conducted with its plans in the subsequent sections.**

7.2 TEST PLANS DERIVED

7.3 RESULTS FROM TESTING

CHAPTER 8. DEPLOYMENT STRATEGY

8.1 VERSION CONTROL AND GIT

In the context of an academic mini project at APJA Kerala Technological University (APJA-KTU) utilizing Scrum as a development model, the importance of version control and Git cannot be overstated. Version control systems (VCS) like Git play a crucial role in managing code changes, facilitating collaboration among team members, and ensuring project integrity throughout the development process.

8.1.1 IMPORTANCE OF VERSION CONTROL AND GIT

1. Collaboration and Coordination

Version control systems enable multiple team members to work concurrently on the same project without overwriting each other's changes. With Git, each team member can create their own branch to develop features or fix bugs independently. This branching strategy allows for seamless integration of different contributions, fostering a collaborative environment essential for Scrum teams.

2. Tracking Changes and History.

Git provides a comprehensive history of changes made to the codebase, allowing team members to see what modifications were made, by whom, and when. This audit trail is invaluable for understanding the evolution of the project and for identifying when issues were introduced. In an academic setting, this transparency helps students learn from their coding practices and understand the impact of their contributions.

3. Reverting Changes

In any development project, mistakes can occur. Version control systems like Git allow teams to revert to previous versions of the codebase easily if a new change introduces a bug or error. This capability is particularly beneficial in an academic project where time constraints may not allow for extensive debugging.

4. Enhancing Code Quality

Using Git encourages a culture of peer review through pull requests, where team members can review each other's code before it is merged into the main branch. This process not only improves code quality but also enhances learning opportunities as students receive feedback on their work.

5. Backup and Recovery

Git serves as a backup system for the project files. Since all changes are stored in a central repository, even if a local machine fails or data is lost, the project can be recovered from the repository. This reliability is crucial in academic projects where data integrity is paramount.

6. Streamlining Development Processes

Version control systems facilitate better planning and management of tasks within Scrum sprints. By linking code changes directly to user stories or tasks in the product backlog, teams can maintain alignment with project goals and track progress effectively.

8.2 RELEASE NOTES

Information on product releases will include:

- Versioning details: v1.0 - Initial release with core functionalities.
- Known issues: Minor bugs in notification system.

CHAPTER 9. SUMMARY

9.1 RETROSPECTIVE NOTES

Summary of lessons learned:

- Need for better communication during sprints.
- Action items: Schedule more frequent check-ins.

APPENDICES

SCREEN SHOT OF ACTUAL UIs

COMPLETE CODE (if applicable)

Glossary of terms used throughout the project will also be included for clarity on specific terminology related to Scrum practices and software development methodologies.

Additional resources may include links to relevant documentation or tools used throughout the project development process.

REFERENCES

(Journal paper)

1. Andrews, G.E and D.Bradley (1972) The Burning Velocity of Methane-Air Mixtures, *Combustion & Flame*, 19, 275-288

3. Bruun,H.H (1976) A Note on Static and Dynamic Calibration of Constant temperature Hot-wire Probes, *J.Fluid.Mech*, **76**, 145-155

(Referenced book)

4. Bradshaw,P. An Introduction to Turbulence and its Measurement, Pergamon Press, 1971.

5. G.O.Young, Synthetic structure of industrial plastics, in *Plastics*, 2nd Ed., Vol.3, J.Peters, Ed. New York: McGraw Hill, 1964, 15-64

(Periodicals)

9. J.U.Duncombe, Infrared navigation – Part I : An Assessment of feasibility, *IEEE Trans. Electron Devices*, Vol. ED-11, No.1, 34-39, Jan 1959

(Reports)

15. E.E.Reber, R.L.Michell and C.J.Carter, Oxygen absorption in the earth's atmosphere, Aerspace Corp., Los Angeles, CA, Tech. Rep. TR-0200 (4230-46)-3, Nov 1988.

(Manuals/Handbooks)

23. Transmission Systems for Communications, 3rd Ed., Western Electric Co., Winston – Salem, NC, 1985, 44-60.

31. Motorola Semiconductor Data Manual, Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.

13. A Handbook of Chemical Engineering, 2nd Ed., Vol. II, G.E.Davis, Davis Bros., Manchester, 1903, 345-356

(papers published in conference or symposium proceedings)

25. Lefebvre,A.H. (1965) Progress and Problems in Gas Turbine Combustion, *10th Symposium (International) on Combustion*, The Combustion Institute, Pittsburg, 1129-1137.

(online books)

7. Jones (1991, May 10), Networks, (2nd Ed.) [Online]. Available: <http://www.atm.com>

(online journals)

11. **R.J.Vidmar**, (1992, Aug.). On the use of atmospheric plasmas as electromagnetic reflectors. IEEE Trans. Plasma Sci. [Online}.21(3), 876-880. Available: <http://www.halcyon.com/pub/journals/21ps03-vidmar>