SOFTWARE ENGINEERING
PROJECT PLAN DOCUMENT SUBMISSION

ANANYA BHAT PES2UG20CS045

AMEY MITTAL PES2UG20CS037

ABHISHEK KUTRE PES2UG20CS012

ARADANA DEIVAM PES2UG20CS063

1)We will be using the Agile model to implement our project.The Agile software development life cycle consists of 6 phases:

1. Concept

The scope of the project is determined in this phase. The product owner will discuss key requirements with a client and prepare documentation to outline them, including what features will be supported and the proposed end results.The requirements are kept to a minimum as they can be added to in later stages. The time and cost of potential projects will also be estimated. This detailed analysis will help them to decide whether or not a project is feasible before commencing work.

2. Inception

The team will create a mock-up of the user interface and build the project architecture. The inception stage involves further input from stakeholders to fully flesh out the requirements on a diagram and determine the product functionality. Regular check-ins will help to ensure that all requirements are built into the design process.

3. Iteration

The iteration phase, also referred to as construction. It tends to be the longest phase as the bulk of the work is carried out here. The developers will work with UX designers to combine all product requirements and customer feedback, turning the

design into code. The goal is to build the bare functionality of the product by the end of the first iteration or sprint. Additional features and tweaks can be added in later iterations.

4. Release

The Agile team members will test the system to ensure the code is clean — if potential bugs or defects are detected, the developers will address them swiftly. User training will also take place during this phase, which will require more documentation.

5. Maintenance

The software will now be fully deployed and made available to customers. During this phase, the software development team will provide ongoing support to keep the system running smoothly and resolve any new bugs. They will also be on hand to offer additional training to users and ensure they know how to use the product. Over time, new iterations can take place to refresh the existing product with upgrades and additional features.

6. Retirement

The software development team will first notify users that the software is being retired. If there is a replacement, the users will be migrated to the new system. Finally, the developers will carry out any remaining end-of-life activities and remove support for the existing software.

The reason why we are using this model is as follows:

1. Reduces Technical Debt
2. Easily and Quickly Adapt to Change
3. Total Alignment and Transparency
4. Agile Software Development and Test Minimize Risk
5. Higher Quality Product
6. Predictable Delivery Dates
7. Better Stakeholder Engagement

8. User-Focused Testing
9. Greater Customer Satisfaction
10. Better Project Control

2: Identify the tools which u want to use throughout the lifecycle like planning tool, design tool, version control, development tool, bug tracking, testing tool.

Planning Tool - Jira, Trello, Lucidchart
Design Tool - Figma,Canva
Version Tool - GitHub
Development Tool - VSCode, Jupyter Notebook
Bug Tracking -Jira
Testing - VSCode, Jupyter Notebook

3: Determine all the deliverables and categorize them as reuse/build components and justify the same.

1) Login/ Signup page with the database integration:
   a) The login page allows the user (teacher or the student) to access the product by entering their username and password
   b) The login page is same for both teacher and student but after logging in they have different interfaces
   c) If a user forgets his/her credentials, then there will be a link to change them
   d) For a new user signup option will be there where all his details with image will be taken by the system to identify him/her
2)
   2. Livestream Camera (Accessible by teachers)
   a) Once the teacher logs in, the livestream camera will be activated and the live input of the classroom will be fed into the camera, by recording a video of all the student's faces in the classroom.
   b) She/He must make sure that all the students are visible and can be detected by the camera.
   c) The faces of the students will be stored in a database and compared with a pre-existing database of the students' images.
   d) If there's a match the student will be marked as present and his/her attendance will be updated.
   e) If a match is not found, then the student will be marked as absent.
3)
   3. Student Interface (after student login)
   a) Student details like his/her name, address, class section and roll number will be displayed
   b) Once the student logs in, their attendance will be displayed in the various subjects.
   c) The student cannot make any changes to the attendance.
   d) Raise Query button where the student can indicate or communicate with the respective authority if he/she was marked wrongly absent.

e) If the attendance is below the minimum required percent in any subject, then a notification or an alert message will be given to the respective student

4)

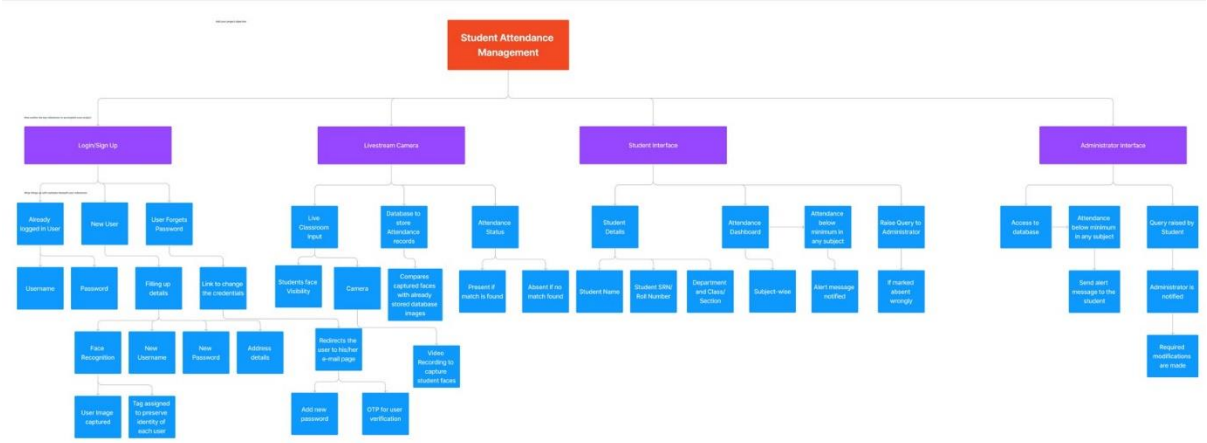
4. Administrator Interface

a) In case the student has been wrongly marked as present, then the administrator has to be contacted

b) The administrator has access to the database. Access is restricted to students and teachers.

c) Once a query request has been raised, the administrator is notified and he re-checks the live video and makes the necessary changes.

4: Create a WBS for the entire functionalities in detail.



5: Do a rough estimate of effort required to accomplish each task in terms of months.

Milestone	Start Date (tentative)	Deadline (tentative)	Member responsible
Login/SignUp	23/9/22	30/9/22 (1 week)	Ananya Bhat Abhishek Kutre
Live Stream Camera with ML implementation	1/10/22	15/10/22 (2 weeks)	Amey Mittal Aradana
Student Interface	16/10/22	23/10/22 (1 week)	Ananya Bhat Abhishek Kutre
Administrator Interface and UI improvisation	24/10/22	7/11/22 (2 weeks)	Amey Mittal Aradana

6: Create the Gantt Chart for scheduling using any tool.