Project Plan

for

PESU APP

Project ID: A16

Prepared by:

Lavitra Kshitij Madan - PES1201800137

Vishnu Charan Golugula - PES1201801230

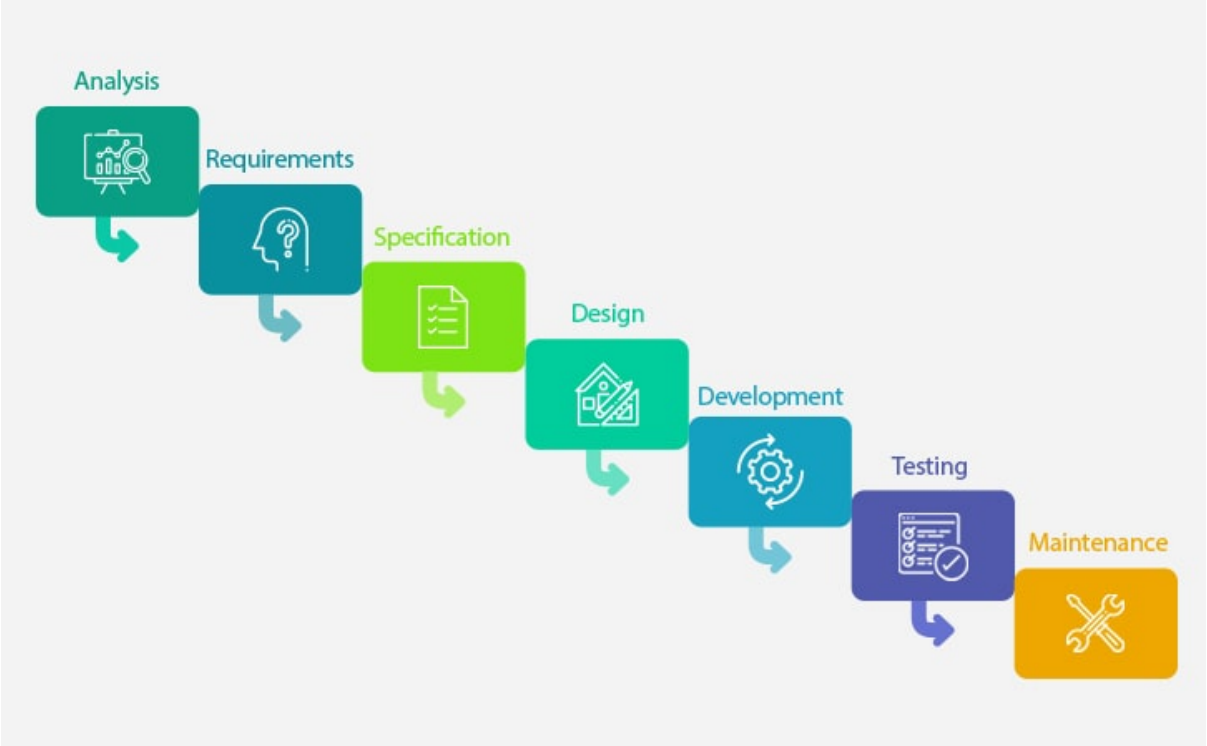
Arjun Chengappa - PES1201800119

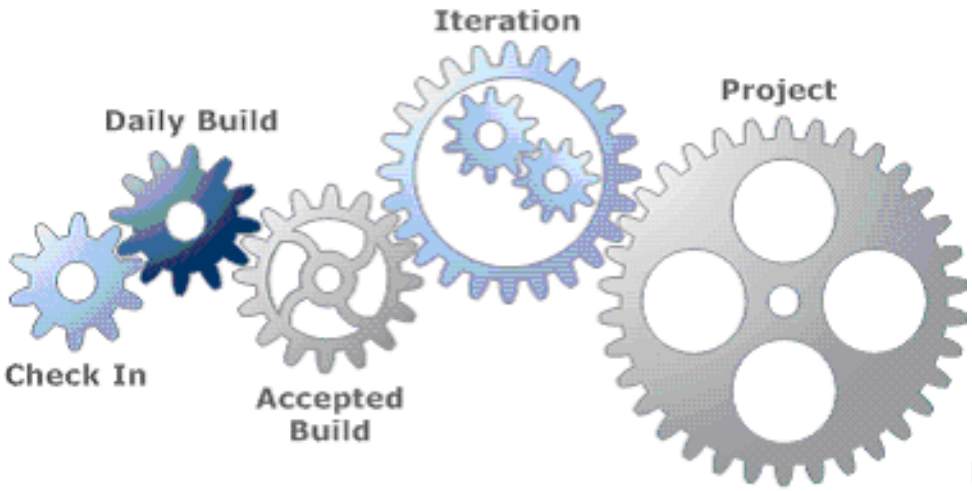
PES University

20 February 2021

Execution lifecycle used:

We have decided to use the Agile Methodology for the given application.





[1]

The steps are:

- ➔ Analysis
- ➔ Requirements
- ➔ Specifications
- ➔ Design
- ➔ Development
- ➔ Testing
- ➔ Maintenance

The justification for choosing the Agile model is as follows:

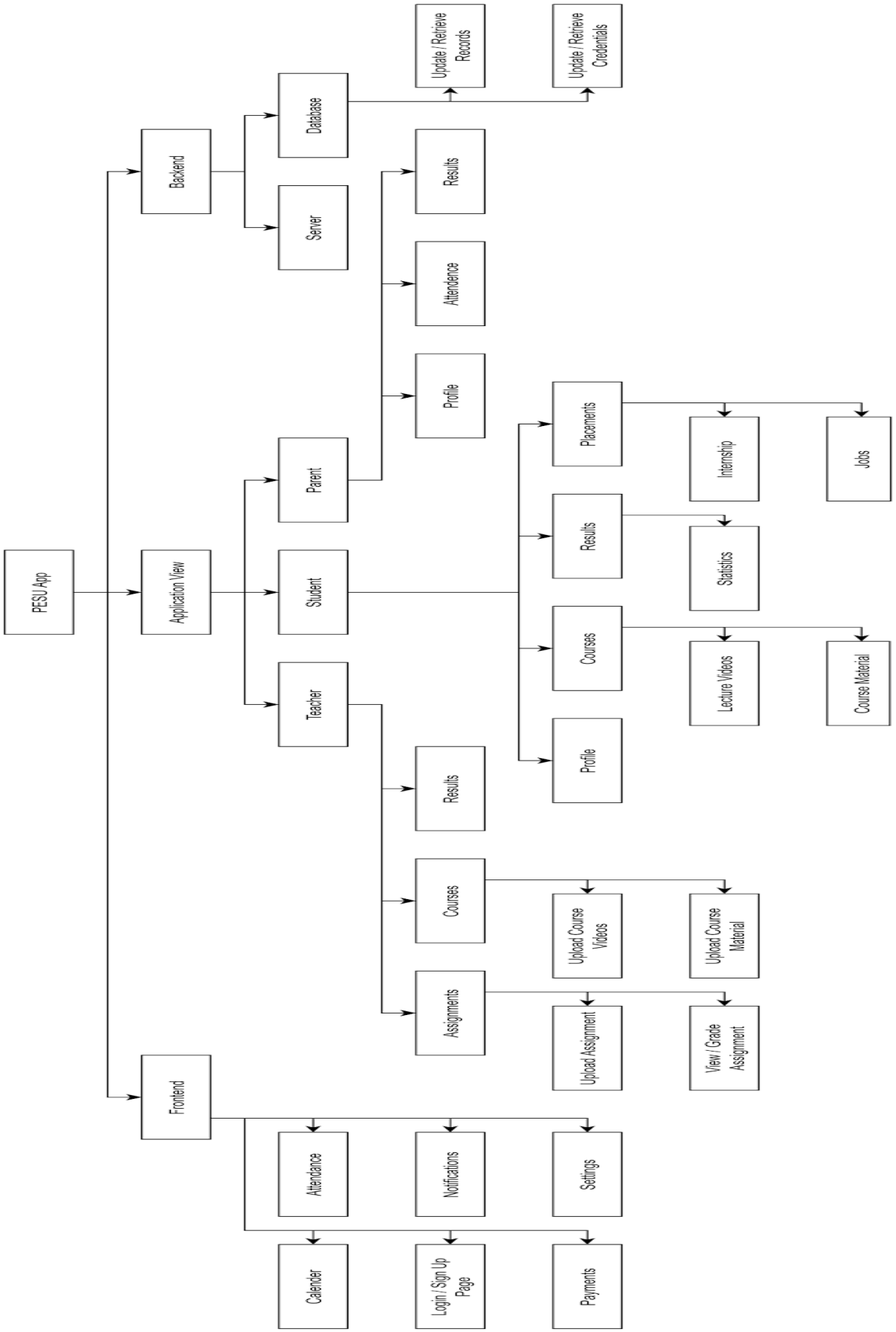
It is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of a working software product. Agile methods break the product into small incremental builds. These builds are provided in iterations.Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features,the final build holds all the features required by the customer. Our application will be developed and rolled-out in phases and will also include new features at every update. So, it involves iteration as well as increments.

Tools used:

- ➔ Planning tools:
  - Google docs
  - Google drawings (Paint)
  - Google sheets (Spreadsheet)
  - Google Keep (ToDo List)
- ➔ Design tools:
  - StarUML
  - Diagrams.net (formerly draw.io)
- ➔ Version control tool:
  - Git version control by GitHub
- ➔ Development tools:
  - HTML, CSS and JavaScript
  - PHP
  - Python 3.6 (Kivy)
  - SQL / SQLAlchemy
  - Visual Studio Code (or Atom)
  - Heroku
- ➔ Bug tracking tool:
  - GitHub Issues
  - sentry.io / pypi.org
- ➔ Testing tool:
  - Appium/TestProject

Work breakdown structure:

https://drive.google.com/file/d/1VPfGjvjUOXGrKRP1Jretxe_MlMMrcTKb/view?usp=sharing - Work Breakdown Structure



Deliverables:

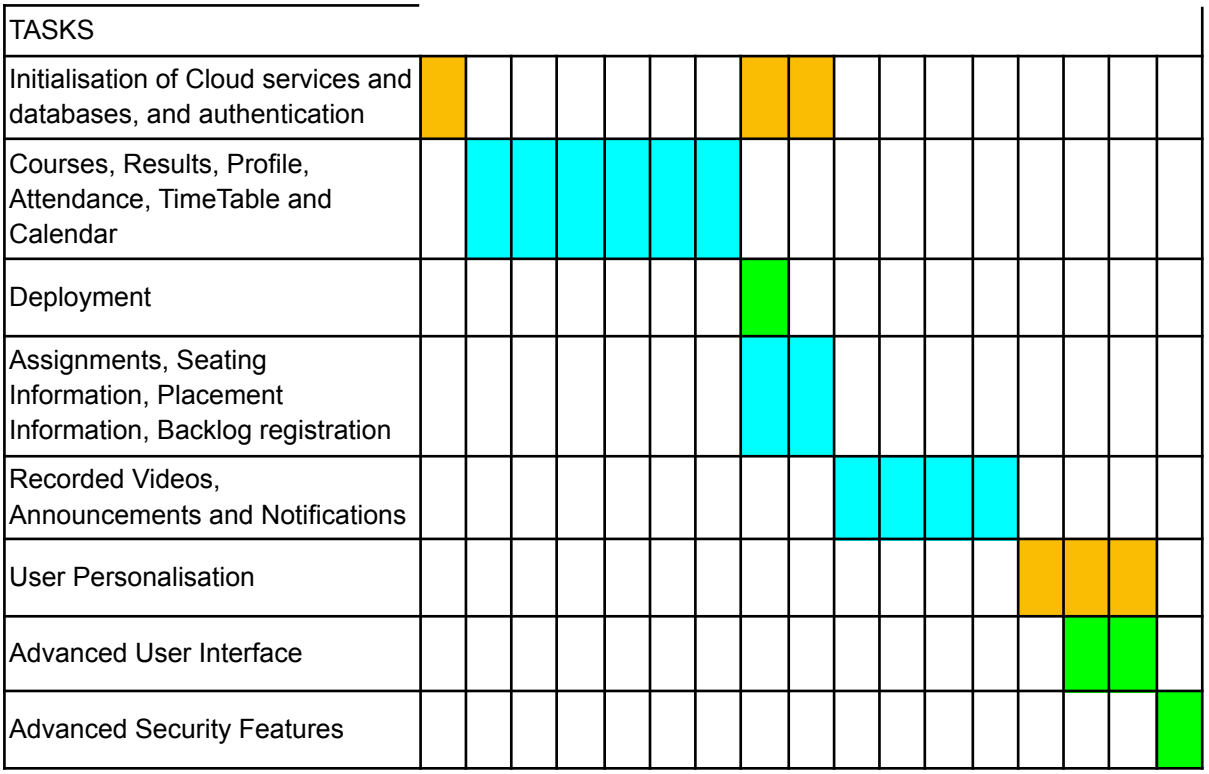
Sl No.	Deliverable	Description	Category
1	Software Requirements Specification (SRS) document	An SRS specifies all the features, requirements and a rough idea of the implementation details of the project. It minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. It sets the tone for the project.	Build : This document is initially written as version 1.0 Reuse : Changes can be made in the later versions to improve based on feedback for our agile implementation.
2	Project Plan document	The project plan outlines all the details of procedures and timelines of execution. Project planning plays an essential role in helping guide stakeholders, sponsors, teams, and the project manager through other project phases. It deals with the SDLC part of the project.	Build : It is created to document the initial plan. Reuse: It can be molded into slightly different versions since we are using agile methodology.
3	Initialisation of Cloud services and databases, and authentication	Databases such as profiles, login credentials, placement information, results, attendance, etc. will be initialised in this task. We will also need cloud storage to store recorded class videos, assignments, etc.	Build : The database and cloud storage is built and initialised for future use.
4	Courses, Results, Profile, Attendance, TimeTable and Calendar	Building the user interface for displaying courses, results, profile, attendance, timetable and calendar. This also involves connecting to their respective databases and cloud storages.	Build : These features are built and implemented in the application.
5	Assignments, Seating Information, Placement Information, Backlog registration	Building more features such as assignments, seating information, placement information and backlog registration sections. Allowing various types of users to have different interfaces and options.	Build : These features are built and implemented in the application.
6	Recorded Videos, Announcements and Notifications	Adding the last phase of features involving recorded videos, announcements and giving the users an option to turn on notifications for the application.	Build : These features are built and implemented in the application.
7	User Personalisation	Giving each user a personal touch by linking results to profile (users will now see only their own results, unlike result display board). Other personalisations like allowing specific notifications, viewing only courses that the user has enrolled into and an option	Build : New features are built on top of the existing ones to give it a personal touch. Reuse : All the old features are reused.

		to continue watching recorded class videos from where they left last time.	
8	Advanced User Interface	Enhancing user interface, making the application look more pulchritudinous and easy to use. Making it require less bandwidth of data/network. Allowing screen rotation also.	Reuse : All the old features are reused to give it a new easy to use and access look.
9	Deployment	Deploying the application and making it open for testing. It will be published once it gets approved and passes all tests. Making the application accessible from any valid device.	Reuse : All the features are deployed and thus being reused by the hosting services.
10	Advanced Security Features	Adding advanced security features to make it free of any loopholes that may lead to attacks by crackers and allowing users to use fingerprint scanner (if available) to view the results, profile and placements page.	Build : New important security features/patches are continuously built and passed as updates. Reuse : Old security features apart from all the other features are also reused.

Rough effort estimate:

Tasks	**Person-Hours**
Software Requirements Specification (SRS) document	0.25
Project Plan document	0.25
Initialisation of Cloud services and databases, and authentication	0.75
Courses, Results, Profile, Attendance, TimeTable and Calendar	1.5
Assignments, Seating Information, Placement Information, Backlog registration	0.5
Recorded Videos, Announcements and Notifications	1
User Personalisation	0.75
Advanced User Interface	0.5
Deployment	0.25
Advanced Security Features	0.25

Gantt chart for scheduling tasks:



References

https://www.projectsmart.co.uk/which-life-cycle-is-best-for-your-project.php

https://medium.com/@melsatar/software-development-life-cycle-models-and-methodologies-297cfe616a3a

https://www.pmclounge.com/types-of-project-management-life-cycle/

https://www.geeksforgeeks.org/software-engineering-cocomo-model/

https://peerbits-wpengine.netdna-ssl.com/wp-content/uploads/2018/10/agile-software-development-waterfall-model-approach.jpg

https://agilemanifesto.org/