SOFTWARE ENGINEERING

HR MANAGEMENT SYSTEM

Team:

Sneha Srinivasan PES2UG20CS342
Srushti V Reddy PES2UG20CS353
Tejal Meda PES2UG20CS369
Vaishali Ganeshkumar PES2UG20CS376

PROJECT PLAN

1: Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model.

The Agile software development life cycle model is used for this project. It is an iterative approach to delivering a project throughout its life cycle. Iterative or agile life cycles are composed of several iterations or incremental steps towards the completion of a project.

Agile is chosen as it provides advantages such as:

- It virtually eliminates the chances of absolute project failure. Working in sprints allows teams to develop a working product from the beginning or fail fast and take another approach.
- Agile builds accountability, encourages diversity of ideas, allows the early release of benefits, and promotes continuous improvement.
- It allows decisions to be tested and rejected early with feedback loops providing benefits that are not as evident in other models.

The Agile software development life cycle consists of six phases.

1. Concept

A product owner determines the scope of their project. 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.

2. Inception

The inception stage involves further input from stakeholders to fully discuss and plan 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

In this phase, 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. This stage enables developers to create working software quickly and make improvements to satisfy the client.

4. Release

The quality assurance team performs tests to ensure the software is fully functional. 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.

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.

6. Retirement

Two reasons why a product enters the retirement phase: either it is being replaced with new software, or the system itself has become obsolete or incompatible with the organization over time. 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.

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

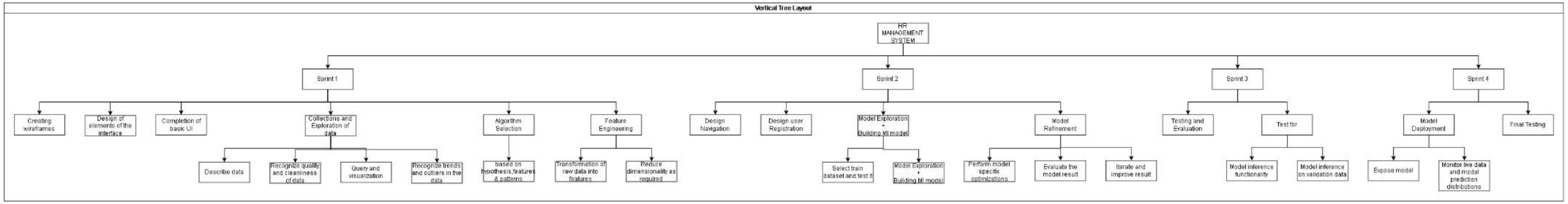
- Planning tool: Jira
- Design tool: Figma
- Version control: Git
- Development tool: HTML, CSS, PHP, Mysql
- Bug tracking tool: Jira, GitHub

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

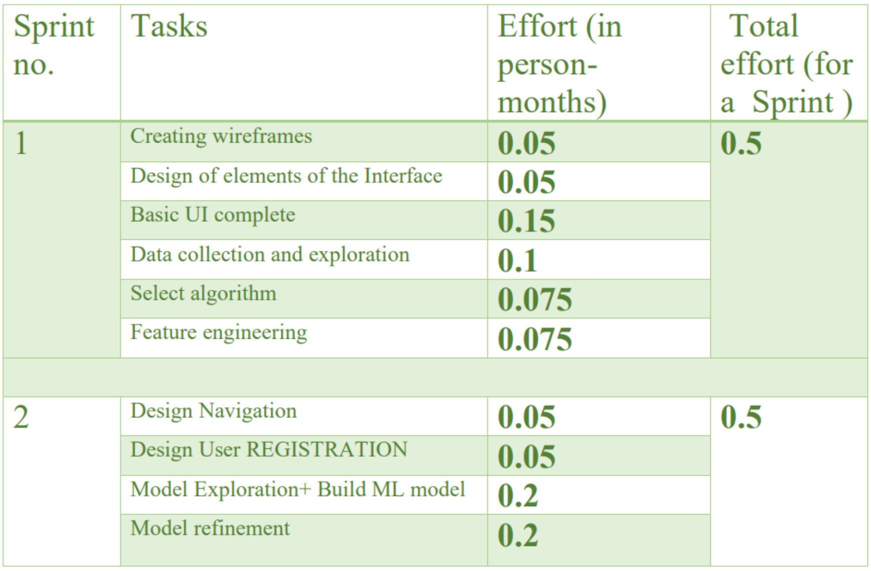
The key deliverables of our project are:

1. User interactive Design (UI) -It can be a reusable model to save the costs in our project and we can add our own features for a unique interface.
2. Preprocessed data – It is a build model since the data required for our database is based on a particular company.
3. Converged model – It is a build model since the process is specific to our model.
4. Optimized model – It is a build model since every project will have different problems and their solutions will be specific to it.
5. Model deployment – It is the process of engineering the task of putting the model to use and hence it is a build model.

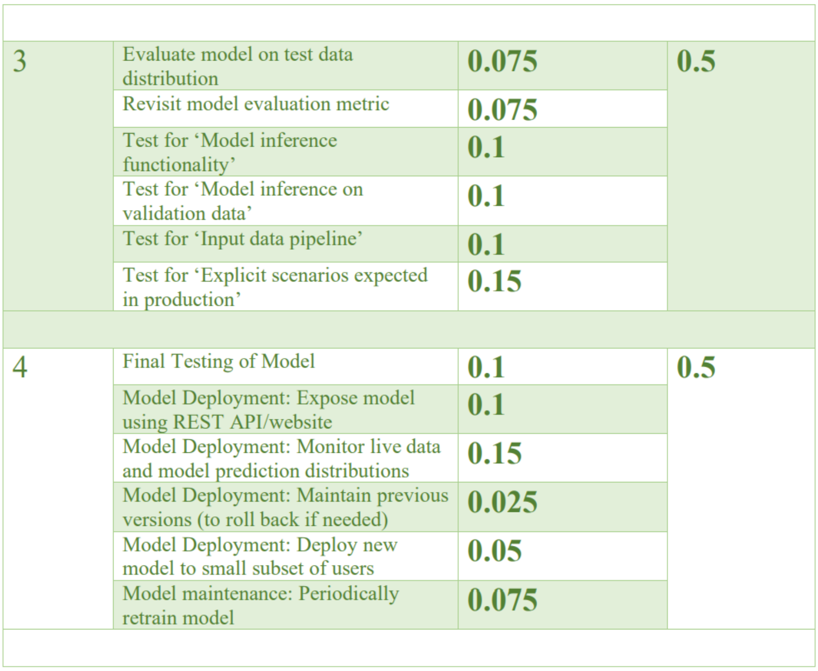
4: Create a WBS for the entire functionalities in detail.


Vertical Tree Layout
HR MANAGEMENT SYSTEM
Sprint 1
Creating wireframes
Design of elements of the interface
Completion of basic UI
Collections and Exploration of data
Describe data
Recognize quality and cleanliness of data
Query and visualization
Recognize trends and outliers in the data
Algorithm Selection
based on hypothesis,features & patterns
Feature Engineering
Transformation of raw data into features
Reduce dimensionality as required
Sprint 2
Design Navigation
Design user Registration
Model Exploration + Building Ml model
Select train dataset and test it
Model Exploration + Building Ml model
Model Refinement
Perform model specific optimizations
Evaluate the model result
Iterate and improve result
Sprint 3
Testing and Evaluation
Test for
Model inference functionality
Model inference on validation data
Sprint 4
Model Deployment
Expose model
Monitor live data and model prediction distributions
Final Testing

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



Sprint no.	Tasks	Effort (in person-months)	Total effort (for a Sprint)
1	Creating wireframes	0.05	0.5
	Design of elements of the Interface	0.05	
	Basic UI complete	0.15	
	Data collection and exploration	0.1	
	Select algorithm	0.075	
	Feature engineering	0.075	
2	Design Navigation	0.05	0.5
	Design User REGISTRATION	0.05	
	Model Exploration+ Build ML model	0.2	
	Model refinement	0.2	



3	Evaluate model on test data distribution	0.075	0.5
	Revisit model evaluation metric	0.075	
	Test for 'Model inference functionality'	0.1	
	Test for 'Model inference on validation data'	0.1	
	Test for 'Input data pipeline'	0.1	
	Test for 'Explicit scenarios expected in production'	0.15	
4	Final Testing of Model	0.1	0.5
	Model Deployment: Expose model using REST API/website	0.1	
	Model Deployment: Monitor live data and model prediction distributions	0.15	
	Model Deployment: Maintain previous versions (to roll back if needed)	0.025	
	Model Deployment: Deploy new model to small subset of users	0.05	
	Model maintenance: Periodically retrain model	0.075	

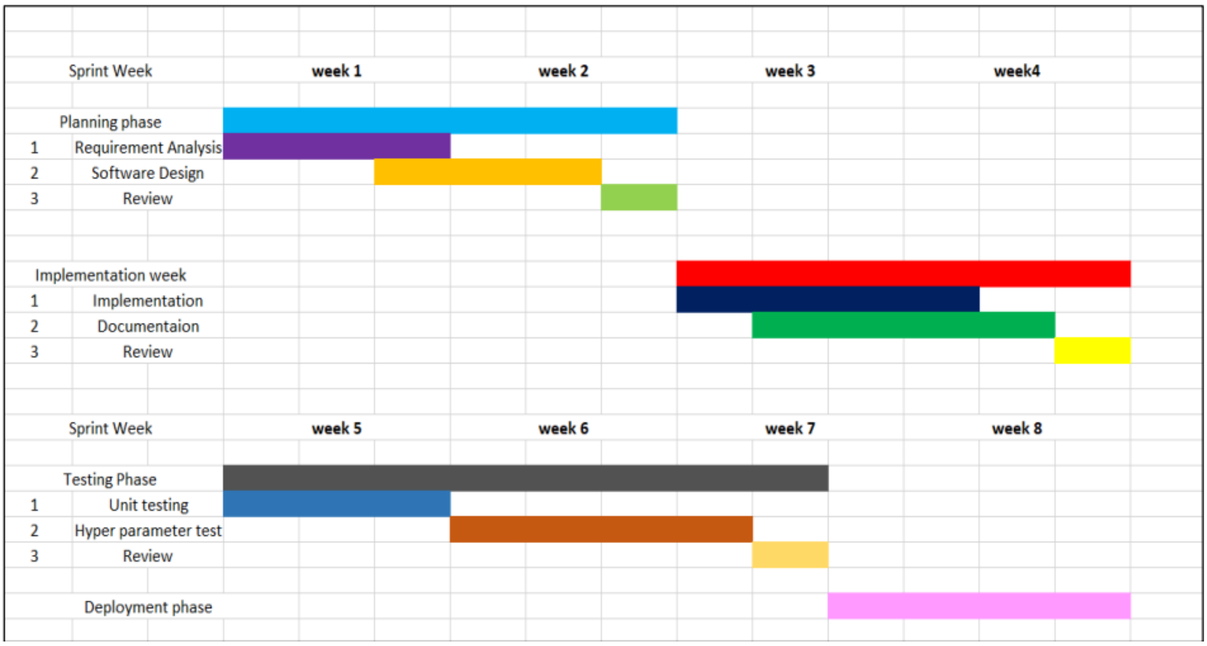
Note:

1) **Note1:What is person-month?**
   Person-month is the measurement unit for effort in software engineering. 1 person-month is the effort put by a person in one month. [Though, 100 person-months does not mean, work

effort put by 100 person in one month or 1 person in 100 months.]

2) **Note2: Estimation Method used here:**
   'Bottom-up estimation' is used for this. [In this, the big and complex tasks are broken down into smaller pieces or tasks. Then, each piece/small task is estimated separately and the values are added up.]
3) **Note3: What is person-hours**?
   Person-hours is the amount of effort required to complete an hour of work by the average worker at the organization. Effort is generally measured in person-hours.

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


Sprint Week
week 1
week 2
week 3
week4
Planning phase
1 Requirement Analysis
2 Software Design
3 Review
Implementation week
1 Implementation
2 Documentaion
3 Review
Sprint Week
week 5
week 6
week 7
week 8
Testing Phase
1 Unit testing
2 Hyper parameter test
3 Review
Deployment phase