Scrumbledore's Army - Phase 1

Driver	APAD Program Director
Approver	Abhay Samant, Evan King
Contributors	@ Aboorva Erode Baskaran @ Amrit Pradhan @ Deeksha Pandit @ Meghna PM @ Sanjo Shaju @ shivangi
Sprint velocity	Weekly - Planning to complete 15 story points in one sprint. Individuals will have <= 3 story points each.
Backlog grooming	Monday - 12:30 pm
Sprint retrospective	Friday - 9 am
Daily scrum	Monday / Wednesday 12:30 pm - 12:45 pm Tuesday / Thursday 9 am - 9:15 am Friday 9 am - 9:15 am Where: Online / Offline
Collaboration tools	Project management: Jira, Confluence / Google docs / Wiki, Github, Mountain goat, Story Point Poker, Figma Team management: Gmeet / Zoom / Slack
Implementation methodolgy	Agile
Confluence page	https://scrumbledoresarmy.atlassian.net/wiki/spaces/ATP/pages/163843/Scrumbledore+s+Army+-+Phase+1
Jira board	https://scrumbledoresarmy.atlassian.net/jira/software/projects/SA/boards/1
Git repo	https://github.com/aboorvaeb/Scrumbledore-army-APAD-Team-Project
Status	IN REVIEW

🚀 Project overview

Goal	Exploring full-stack application development as part of Advanced Programming and Application Development course
Opportunities	Explore front-end and back-end tools for application development and project management tools for driving a full-stack project

Challenges	Visualising and developing under the stipulated time-constraint
Competitive environment	Emerging competitive players in the Hardware-as-a-service vertical
Selling points	Managing resource optimisation for Hardware resources off the shelf as a cloud service offering for users
Value propositions	Provision of Hardware-as-a-service offering in the market in tune with the current SaaS offerings

🏆 Objectives

Scrumbledore's Army has envisaged offering a hardware optimization model as a cloud offering to users so that the users can effectively manage hardware resources as a project, which can be mapped to them and worked in a decentralized manner, which takes into account conflict management and other aspects of the project.

Objective	Priority
User Management	HIGH
Project Management	HIGH
Resource Management	HIGH
Conflict Management	LOW

✨ Intended Audience

The project is intended to cater to the following audience -

- End Users - for using the resource optimization model for their respective projects
- Administrator - for managing resources, projects, and users overall

🎨 Key requirements

- User Management - To provide options for end users to create their User credentials, login/logout of the application, search for their respective projects or create projects. In addition, users should be able to add resources to the pool or withdraw resources from the pool.
- Project Management - To provide an option for end users to create Projects mapped to them and in turn handle resource sets within each project
- Resource Management - To ensure Resource pool with Capacity, Availability, and provision of option to enter resources that need to be added or withdrawn to the pool
- Conflict Management - To eliminate concurrent requests to hardware resources by users. This can be extended to user creation too in case credentials conflict among multiple users.

🚚 Deliverables

Deliverable	Channel
Wiki Page	Jira / Confluence
Project Design & Methodology	Jira / Confluence
Development App	Github
Design Document	Figma

💰 Budget

Cost - NIL for the pilot model

Time - 21 days

🕛 Timeline & review process

Deliverables due date	**22 July - Project Methodology Plan** **05 July - Final App Development** **11 July - Potential Capabilities**
Approval process	Review 1 - 23 July Review 2 - 06 July
Creative reviewers	Abhay Samant, Evan King

🎨 Assumptions & Dependencies

- The above plan is tentative and may change based on the detailed requirements to be gathered as part of the Discovery phase.
- The actual start date and end date of the project are directly dependent on the Program's actual commencing date.
- Sign-off of the solution document (BRD) is a critical milestone to begin the design & implementation phase.
- The timeline and product details may be revisited during the design phase.
- Any change or addition to the scope mentioned in this document shall directly impact the implementation timeline, along with the cost.
- The overall implementation shall be governed by a robust Governance Model including all stakeholders.
- All third-party systems integration with the core system will be based on REST APIs / web services and the REST APIs availability is critical for integration.
- The program team should be empowered to take decisions required during program execution
- Data to be made available in the format defined
- Any changes to the third-party applications or custom development on third-party applications are out of scope in the current phase and are to be used for the Scalability document in the last phase.

🚚 Tools Adopted

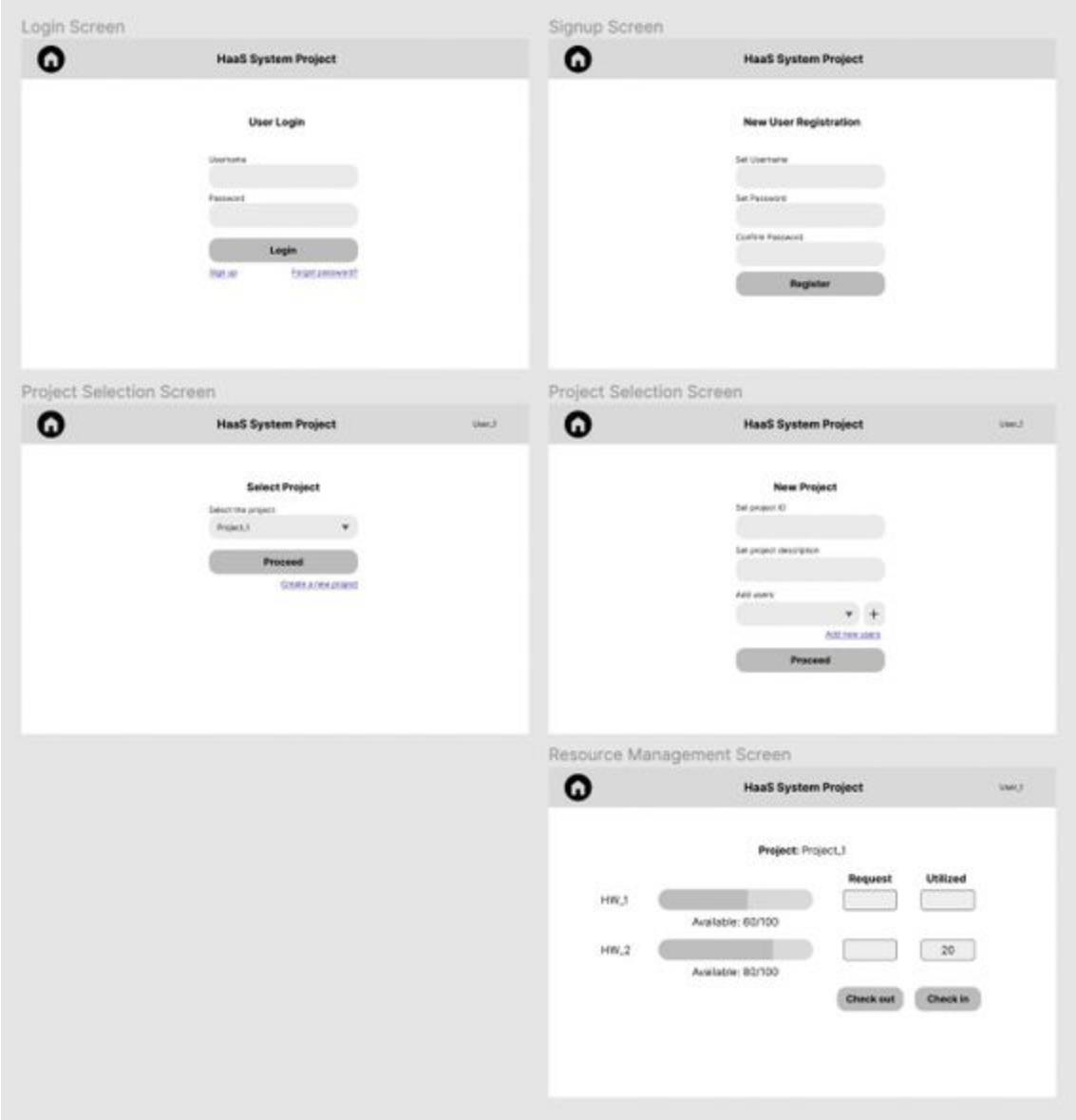
- React (on VS Code Editor) - Front-end application development
- Python (on VS Code Editor) - Back-end core application development
- Flask - Back-end micro-framework based on Python
- MongoDB - SQL-based Database applications
- Heroku - Cloud deployment of application

💻 Approach

- Concurrent development on individual front-end and back-end application components
- Connection of back-end to database
- Integration of front-end and back-end components using APIs
- Deployment of back-end and database components on Heroku as cloud deployment

📈 Prototype

Link to interactive model: Website wireframe (FIGMA)



✨ High-level Architecture Diagram

Below is a high-level architecture diagram for the working application -