SCHOOL OF TECHNOLOGY

BEACHELOR OF SCIENCE IN SOFTWARE DEVELOPMENT

UNIT CODE: BSD 3106

ADMISSION: 21/04706

NAME: DUNCAN MUNENE MURIUKI

NAME OF THE SUPERVISOR: WINNIE OKONG'O

PROJECT TITLE: RESOURCE SCHEDULING AND OPTIMIZATION WEB APP

SYSTEM REQUIREMENT SPECIFICATION

INTRODUCTION

1.1 Purpose

The SRS for the Resource Scheduling and Optimization Web Application is a very important document that clearly explains the objectives, features, and limitations of the system. This document acts like a bridge between stakeholders and the development team so that everyone speaks the same language with regard to the system's capabilities and what it would serve. A clarity of such magnitude greases the skids for design, development, testing, and deployment efforts while keeping the application in tune with the needs of the user and aligning it with the overall project objectives.

The SRS helps in decreasing risks, avoiding costly mistakes, and overcoming the actual time challenges in scheduling by properly defining the purpose, scope, and requirements of the system. Both the functional and non-functional requirements of the system are explained here, as well as the constraints and dependencies, for thorough comprehension and as a reference throughout the project lifecycle. This SRS would finally help in efficient resource scheduling and optimization of a variety of projects and ensure successful delivery in complex project environments.

1.2 Intended Audience and Reading Suggestions

Target audiences for this document include the following: stakeholders who develop, utilize, and manage the Resource Scheduling and Optimization Web Application. The listed team sets up project managers, resource planners, system developers, and other decision-makers. Each group will apply this document in their own way with the intention of having requirements clearly understood and being effectively put into practice.

Project Managers/ Resource Planners: They will refer to this document to verify whether the system satisfies their scheduling and optimization needs. They will look through the requirements to ascertain that the software covers key needs, such as project scheduling, resource allocation, and optimization.

System Developers: It is the prime authority to design the application and implement the system. It will describe the functional and non-functional requirements in a coherent and consistent way to the developers so that the system is built to achieve the specified objectives.

There are end users, such as project teams or organizations that will use the software to schedule and perform optimizations. They may also go through this document to confirm that the required features are needed, provide clarity on ambiguous features, or even suggest other features that could make the system more user-friendly and efficient.

Testers: The QA testers will use this document to ensure the system is working as expected in terms of functionality and performance during the testing and validation phases.

1.3 Project Scope

The Resource Scheduling and Optimization Web Application is designed to help any organization toward the most efficient management and optimization of its resources for various projects. The system will provide scheduling, resource tracking, and optimization in

such a way that projects are completed within time and budget constraints and maximize the utilization of resources. Features described below are for various types of users:

Features - Administrator

i. User Management: A user administrator can create new users, amend user information at any time, and de-activate/delete accounts if and when necessary.
ii. Project and Resource Management: An administrator can add projects, assign and manage projects, resources, and teams. This would include assigning project deadlines, reserving particular tasks for available resources, and monitoring work currently in progress.
iii. The Optimization settings enable administrators to set up various optimization algorithms, their priorities, constraints, and criteria related to resource scheduling and optimization.
iv. Reporting and Analytics: This provides admins with the facility to report on resource allocation, project progress, and optimizations' performance against goals set by the organization.
v. Permissions and Access Control: The admin can provide access to users for various functions through permission allotment to members of the team, assuring that sensitive data will only be available to those users who have a need-to-know basis.

Project Manager/Planner Features:

i. Setup and Scheduling: This will enable the project managers to set up new projects, plan timelines, and thereby commit resources for the same. Provide the system with tools to create task dependencies and adjust the schedule dynamically based on the availability of resources.
ii. Resource Availability Monitoring: The project manager can go through resource availability, including people and equipment, and even budget constraints, to put in the right resources in the right jobs.
iii. Optimizing Resources: Project managers can apply in-built algorithms that minimize idle time and make maximum utilization of resources.
iv. Overcoming Conflicts: Scheduling conflicts and overloads will be notified to the project managers, who then are allowed to make adjustments.
v. Smoothening and Progress Monitoring: There will be an opportunity to monitor in real time the progress at which a particular project is going on, and based on this, the project manager may make readjustments in schedules or resources when necessary.

Features End User:

i. Registration and Profile Management: The user can register them self and maintain his profile, both personal and organizational, which will store their preferences regarding system notifications and updates.
ii. Resource and Task View: The user can see the tasks assigned to him and the availability of resources and can collaborate with other team members to enable the completion of these tasks.
iii. Task Updates: Users can update the status of their tasks in real-time to ensure that the project managers have updated information to track activities.

iv. Notifications and Alerts: Users will be notified of task assignments and deadlines; any adjustments in the schedule will after all keep them current.
v. Role-Based Self-Optimization Suggestions: The user will be presented with optimization suggestions, relevant to their role in a system, for example, how one can better utilize his time amongst tasks or to which tasks of low priority the resources should be reassigned.

System Features:

i. Automatic Resource Optimization: It will come equipped with advanced algorithms that auto-suggest the most optimal assignment of resources against pre-defined constraints, including time, cost, and resource availability.
ii. Dynamic Scheduling: This will include the ability of the system to update schedules in real time and rebalance resources against project needs, should these change or other unforeseen events arise in the ordinary course. iii. Data Integration and Reporting: The reports provided by the system on project status, resource usage, and efficiency of optimization will be detailed and exportable in various formats, including PDF and Excel.
iii. Scalability: The system will be scalable for any team size-from small groups to large organizations running multiple projects simultaneously that are complex in nature and require a host of resources.

1.4 References

1. Pinedo, M. L. (2016). Scheduling: Theory, Algorithms, and Systems (5th ed.). Springer.

2. Taha, H. A. (2017). Operations Research: An Introduction (10th ed.). Pearson.

3. Garey, M. R., & Johnson, D. S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman.

4. Silver, E. A., Pyke, D. F., & Thomas, D. J. (2016). Inventory and Production Management in Supply Chains (4th ed.). CRC Press.

5. Pinedo, M., & Chao, X. (1999). Operations Scheduling with Applications in Manufacturing and Services. McGraw-Hill.

6. Kis, T. (2014). "Job Shop Scheduling with Resource Constraints." European Journal of Operational Research, 236(1), 64-72.

7. Kolisch, R., & Hartmann, S. (2006). "Heuristic Algorithms for Resource-Constrained Project Scheduling: Classification and Computational Analysis. "Project Management Journal, 37(5), 21-37.

8. Drezet, L. E., & Billaut, J. C. (2008). "A Project Scheduling Problem with Resource Constraints and Precedence Relations." Journal of Scheduling, 11(4), 241-256.

9. Vanderbei, R. J. (2014). Linear Programming: Foundations and Extensions (4th ed.). Springer.

10. Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., & Shmoys, D. B. (1993). The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. Wiley.

Overall Description

2.1 Product Perspective

The Resource Scheduling and Optimization Web App is a standalone solution that handles project scheduling and resource allocation in highly complex environments. In contrast to other systems, which are part of a greater family of products, this web application is tailor-made from scratch for focused use by project managers and team leaders who want to simplify resource management processes. It minimizes resource wastage and optimizes project timelines while ensuring effective task allocation.

It should be a unified platform of the system for resource management, schedule tracking, and reporting to enhance decision-making and performance concerning various projects. It will enable the assignment of resources to tasks, understand the status of execution of projects, and find out bottlenecks, if any. Besides that, the application would be integrated with optimization models which would provide the optimal resource allocation strategies concerning constraints and requirements of the project at hand.

The Resource Scheduling and Optimization Web App interacts with several stakeholders and integrations from other external systems to work smoothly. Thus, important parts of the entire system are:

i. Project Managers: These personnel are responsible for managing activities, resources, and schedules related to various projects.
ii. Team Members: Those team members who receive task and resource assignments provided by project managers, and who participate in the execution of the plan.
iii. Optimization Engine: An internal engine that will review resource allocations done by project managers and project timelines with the purpose of recommending the best strategy of scheduling.
iv. Notification System: This module will provide alerts, updates, and reminders to users relating to project status and task deadlines.
v. Database Server: This is the central repository that will store all the data related to projects handled or to be handled, like resource allocation, project timelines, user information, and optimization outputs.

This web-based application is designed to enhance project management efficiency through automation in scheduling, optimizing resource utility, and the presentation of factual insights to the users.

2.2Product Features

User Registration:

i. Personal information and user registration with the login history.
ii. Secure authentication mechanisms needed for access to protect the system.

Project Management:

i. Develop work packages, manage, and command issues in several projects.
ii. Setup of deadlines for projects, and resource allocation and monitoring project result expansion.

Resource Allocation:

i. Resource compatibility according to its needs and project schedule optimization.
ii. Minimization of the priority schedule through optimization engine inbuilt has been received automatically.

Scheduling Tasks:

i. Algorithmic scheduling of project time-dependent tasks and resource availability.
ii. Manual actions in cases that the main system auto assign the tasks.

User and Role Management

Admin-based functions help to create and manage project manager, team member, and sysadmin roles in the system.

Based on the different rights that are provided to the users are drilled through the role-based access control (RBAC).

Notification and Alerts

i. Auto-email/SMS alerts of the assignment of tasks, deadlines, and updates.
ii. In-app push notifications about the project critical updates and reminders.

Reporting and Analytics:

i. Efficiency, project timeline, and performance report strategy are the key requirements.
ii. Figural statistics to track the efficiency of resource allocation.

Integration with External Systems:

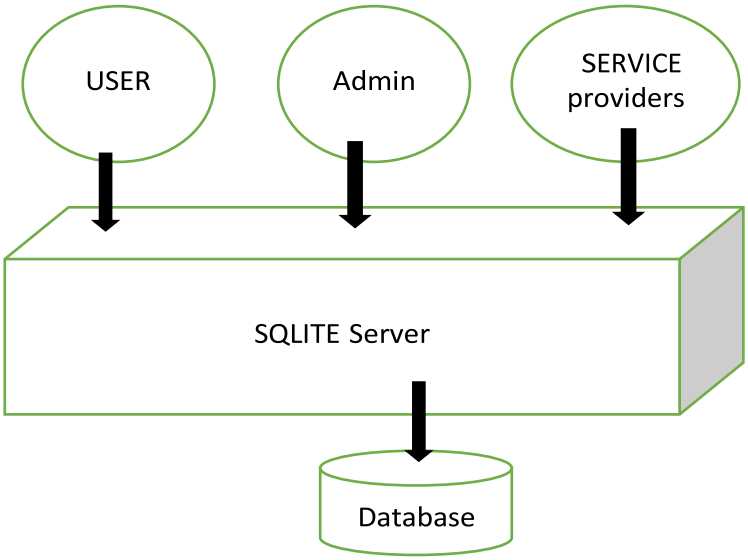
i. Integration with third-party calendars, project management tools, and databases for proper data synchronization.

Admin Dashboard:

ii. Admin interface to all the projects, resources, user accounts, and system settings.
iii. Role-Based Access Control: The administrators are provided with a great deal of access that they can use to extend their skills and thus manage the system on a larger scale.

Customer Support:

i. Support the feature in such a way that users can conveniently ask questions related to the system or report any issues.
ii. The support staff can send the user a message via the in-app messaging service or even email and subsequently the support.



2.3 User Classes and Characteristics

Access to the Resource Scheduling and Optimization Web App user classes has been envisaged. However, the majority of the user communities include a hierarchy of the characteristics, access, as well as levels of expertise. To that effect, to cover a variety of user classes most will include:

1. System Administrators

Characteristics:

i. Frequency of Use: Daily or as need for system management tasks.
ii. Subset of Functions: It would have complete access to all the functions of the system, including user management, project configuration, and resource optimization settings.
iii. Technical Expertise: High technical expertise, system administration, and security management.
iv. Security/Privilege Levels: Highest privilege levels in accessing all modules within the system, configuration settings, and user roles.
v. Educational Level/Experience: Probably attains a background in IT, system administration, or computer science with extensive experience in managing web-based applications.
vi. It is responsible for managing system operation, user management, resource optimization rules, workflow settings configuration at the project level, and security settings.

2. Project Managers

Characteristics:

i. Frequency of Use: Daily to weekly depending on the number of projects at hand.
ii. Subset of Functions: Can create, manage, and assign tasks and resources to team members; can also view resource utilization reports and analytics.

iii. Technical Expertise: Average to above-average technical expertise in the usage of project management tools; not necessarily in coding and/or system administration.
iv. Security/Privilege Levels: Medium-level access, typically restricted to functions related to project management: e.g., task scheduling, resource allocation, reporting.
v. Level of Education/Experience: A degree in Business Administration or Management or Engineering-related fields, with experience in project scheduling and resource management.

Responsibilities: Project Management- resource allocation, implementing projects on time, and using the best approach towards task scheduling. Requires real-time data to decide on the allocation of tasks and how teams should be managed.

3. Resource Managers

Characteristics:

i. Frequency of Use: Daily or as dictated by the project activities.
ii. Subset of Functions: It will be mainly concerned with resource availability, keeping a tab on it, and putting it to judicious use. They will also receive notifications regarding resource availability and usage.
iii. Technical Expertise: Average to High level of technical expertise, primarily related to resource management and optimization tools.
iv. Security/Privilege Levels: Access shall be restricted to resource management only, as they will have access to reports regarding resource usage and allocation.
v. Education Level/Experience: Operations management, engineering, or logistics background required; specialized training or experience in resource management and optimization.

Primary Responsibilities: Provide optimum resources allocation across projects and tasks; resolve bottlenecks; report the status of resource availability to project managers.

4. Team Members/Employees

Characteristics:

i. Frequency of Use: Moderate to daily depending on the schedule of projects and tasks assigned.
ii. Subset of Functions: Access to the assigned tasks, status of tasks, and deadlines. Limited reschedule or update progress.
iii. Basic level of technical expertise; use web applications to manage tasks.
iv. Security/Privilege Level: Very limited, having access only to the tasks assigned to them without having any schedule modification privileges or access to system-wide data.
v. Education Level/Experience: The person will likely have an educational background in the relevant domain of engineering, design, and development with basic experience in using task management tools.

Responsibilities: To complete assigned tasks on time, update task progress, and ensure that the tasks they are responsible for meet deadlines.

5. External Stakeholders/Clients

Attributes:

i. Frequency of Use: Infrequent or as required by specific project phases or deliverables.
ii. Subset of Functions: View-only functions to project status, timeline, and progress reports. They can also receive notifications and alerts on updates regarding the projects.
iii. Technical Expertise: Low to average technical expertise, depending on the client's experience with Project Management Systems.
iv. Security/Privilege Levels: Very restricted access - read access to project deliverables and progress information only.
v. Educational Level/Experience: Usually business-oriented or representatives from the client side with little or no technical experience but with profound knowledge about their industry.

Roles: Maintenance regarding the development progress and giving feedback or approval on the project progress.

6. System End Users (Users of the Resource Scheduling Feature)

Characteristics:

i. Frequency of Use: From regular down to occasional, depending on whether team members or external partners have to schedule anything.
ii. Subset of Functions: Ability to view schedules, book resources, and track progress.
iii. Technical Expertise: Low to moderate; users are generally nontechnical, with variable familiarity with online scheduling applications.
iv. Security/Privilege Levels: Basic access to their own schedules; book or modify resource requests in line with business requirements.
v. Educational Level/Experience: Generally varied but very often field workers, contractors, or external partners without technical acumen related to system operation.

Responsibilities: Ensuring that their tasks or projects are properly scheduled and resources available at the required times.

Favoured User Classes

Project managers and resource managers are the preferred classes, since they directly interact with core features of the system: task scheduling, resource allocation, and optimization. The efficient use of the system by them is of utmost importance for success in resource scheduling and optimization efforts.

Less Important User Classes

He is the external stakeholder/client who interacts least with the core feature set of the system. He needs only to see the progress reports and timelines without being an influencing force in

the actual resource scheduling or optimization processes. However, his feedback would not be indifferent in system improvement and alignment towards business goals.

2.4 Operating Environment

The Resource Scheduling and Optimization Web App will be a cloud-based solution and will be accessible via modern web browsers. Following is the breakdown of the basic components:

Hardware Platform: The system will be deployed on regular infrastructure for web servers, scalable resources on the cloud platforms like AWS, Azure, or Google Cloud based on the project scale and user load.

1.Operating System: Application shall be platform independent, accessed via web browsers on multiple operating systems following:

i. Windows 10/11
ii. macOS, latest versions
iii. Linux, including Ubuntu and CentOS

2. Web Browsers: Application is supposed to work in the following browsers' latest versions:

i. Google Chrome
ii. Mozilla Firefox
iii. Safari
iv. Microsoft Edge

3. Back-end Environment:

i. Web Server: Apache or Nginx
ii. Back-end Framework: Django Python-based
iii. Database: PostgreSQL or MySQL

4. Front-end

i. Framework: ReactJS or Angular to provide dynamic UI
ii. Styling: Bootstrap 5 or Tailwind CSS

Third-party Integrations:

i. APIs for Resource Scheduling: It has to be integrated with third-party APIs in respect of synchronisation of the calendar, email/SMS notifications.
ii. AWS S3 to store reports or documents related to scheduling. This environment will make sure that the app is going to operate efficiently, very seamlessly on varied platforms, and be scalable for future growth.

2.5Design and Implementation Constraints

The following constraints will be applicable in the development of the Resource Scheduling and Optimization Web Application:

1. Regulatory and Compliance Constraints:

i. The application will handle user data in compliance with GDPR, General Data Protection Regulation, for privacy and protection.
ii. The secure handling of payment data according to PCI-DSS.

2. Hardware and Performance Constraints:

i. The system should be optimized to run on anything from personal laptops to cloud-hosted environments.
ii. Memory and CPU usage should be at a minimum to scale and perform well in low-resource and high-resource environments.

3. Technological Constraints

i. The system should be done using Django as the backend and Python 3.8 or above as the development language.
ii. PostgreSQL or MySQL - both should be supported.
iii. Frontend: React.js or Vue.js to provide runtime dynamic user interfaces, and Bootstrap 5
iv. Redis for caching; Elasticsearch may be integrated for search functionality.
v. RESTful APIs for communication between services/ components.

4. External Interfaces:

i. Integration with payment gateways like Stripe, PayPal, and M-Pesa for transaction processing; SMS services like Twilio or Nexmo for notifications; and email services like SendGrid or Amazon SES for alerts.
ii. Cloud Services: Hosting and data storage on AWS or Google Cloud; reliance on cloud-specific services such as S3 for file storage and EC2 for hosting.

5. Security Constraints:

i. Secure authentication of users with OAuth 2.0 or JWT.
ii. All sensitive data to be encrypted, while in motion it should be with HTTPS and at rest.
iii. Follow security best practices: prevent SQL injections, XSS protection, and secure payment processing among others.

6. Design Constraints:

i. It needs to be responsive and optimized for desktop and mobile devices.
ii. Design all interfaces to be user-friendly, with intuitive navigation and ease of use for different types of users.
iii. Standardization of UI/UX design, consistent displays, including accessibility guidelines such as WCAG 2.0.

7. Development and Deployment Constraints:

The application deployment will be automated, using pipelines either with the aid of Jenkins or via GitHub Actions.

Some lines for a more general guideline on version control and coding standards were added, as follows:

i. Version control should be done via Git; repositories should be hosted on GitHub or GitLab.
ii. The code should follow PEP 8 for Python and ESLint for JavaScript to maintain the readability and quality of the code.

8. Time Constraints:

i. The development must be done within the agreed-upon timeframe, using timely milestones for every phase in the project, including design, implementation, testing, and deployment.
ii. These constraints will guide the design, development, and deployment processes to ensure the system can fulfill functional or nonfunctional requirements, respectively.

2.6 User Documentation

The following components of user documentation shall be provided with the Resource Scheduling and Optimization Web Application:

1. User Manual:

i. All detailed instructions on registration, logging in, and account management.
ii. Step-by-step explanation of the Scheduling and Optimization features.
iii. How to search for services, book services, and manage services by end users and admins.

2. Online Help:

i. Context-sensitive help available in the web application interface for quick user reference.
ii. FAQs for frequent user queries.

3. Tutorials:

i. Step-by-step video tutorials or guides that explain core functionalities, setting up user accounts, task scheduling, and optimization feature usage.

4. Documentation Format:

ii. HTML-based online help.
iii. PDF format for offline user manuals.
iv. Tutorials and documentation will also be provided through the help section in the application.

2.7 Assumptions and Dependencies

Assumptions:

1. Third Party Service:

i. For different third-party payment gateways like Stripe, PayPal, and M-Pesa, an appropriate API should be available.
ii. Third-party services for SMS and email notifications will be integrated into the system, including but not limited to Twilio and SendGrid.

2. Software at Another Location:

i. It is assumed that Django will be present for the back-end development of the application and React.js or Vue.js will be there at the front end.
ii. The database used could either be PostgreSQL or MySQL, and already setting up access to that is assumed.

3. Cloud Infrastructure:

The project assumes that cloud hosting - AWS, Google Cloud - is used for deployment and that costs are assumed to be stable and uptimes also stable.

4. User Engagement:

The devices of the end users-admin, service providers, customers-are capable of running modern web browsers to access the system: Chrome, Firefox, Safari.

Dependencies:

1. External Libraries and Frameworks:

The system used will be based on Django to handle the backend, while the front will be handled by React/Vue. The database will be on PostgreSQL/MySQL.

The performance and functionality of the app rely a lot on third-party APIs that will handle the payment and notification sections.

2. Cloud Services:

The following cloud providers will be used: AWS/Google Cloud for hosting, S3/Google Cloud Storage for storing.

3. External Authentication:

If the social login features are enabled, authentication relies on third-party OAuth providers like Google or Facebook.

4. Security Compliance

The project's security and data handling are subject to compliance with GDPR, PCI-DSS, and other standards that are supposed to exist for regulatory purposes in the interest of users.

These assumptions and dependencies must be further scrutinized and updated throughout the development process so that successful implementation of the system can be ensured.

3. System Features

3.1 User Registration and Authentication

3.1.1 Description and Priority

Description: The system allows this feature to enable the user to register, authenticate, and thereby access the platform. It manages both users and administrators, hence it should have wide-ranging accesses for each class. A user can create an account by submitting personal details and credentials necessary for account creation.

3.1.2 Stimulus/Response Sequences

- Stimulus: The user clicks on the "Register" button on the homepage.
- System Response: The system prompts for personal details-name, email, and password-and credentials.
- Stimulus: The user submits a registration form with valid details.
- System Response: The system creates a user account and sends a confirmation email for verification of the account.
- Stimulus: The user logs in by entering credentials.
- System Response: The system checks for the login credentials and logs in the user based on user type-admin or user.
- Stimulus: The user provides wrong login details.
- System Response: It will notify the user through the system by showing an error message which highlights the invalid login credentials.

3.1.3 Functional Requirements

- REQ-1: System will provide registration of users based on name, email, and password.
- REQ-2: Following this step, the system should authenticate the users by verifying their credentials and allowing them access into their respective dashboards.
- REQ-3: The system shall securely store passwords using an encryption algorithm like bcrypt.
- REQ-4: The system shall send a verification email on successful registration.

3.2 Subscription Management

3.2.1 Description and Priority

-Description: The feature allows the user to choose and manage their subscription plan between free, monthly, or yearly tier. This feature will also integrate payment processing and access control based on active subscription.

3.2.2 Stimulus/Response Sequences

- Stimulus: The user clicks on the "Subscribe" button; he is interested in subscription plans.
- System Response: The system pops up with free, monthly, and yearly available plans, giving a short description along with pricing.
- Stimulus: User chooses the subscription plan and pays for the same by providing the details.
- System Response: It will process the payment information provided and changes the subscription status of the user.
- Stimulus: User tries to access the premium services without an active subscription.
- System Response: The system does not allow entry into premium services and prompts the user to upgrade the subscription.

3.2.3 Functional Requirements

- REQ-1: The system shall provide the user with a selection choice for free, monthly, and yearly subscription plans.
- REQ-2: Payment processing shall be done by the system through any one of the external payment gateways such as Stripe.
- REQ-3: The system shall grant access to only premium features available to users, according to their subscribed package that is currently active.
- REQ-4: The system shall notify the users when their subscription date is near expiration.

3.3 Dashboard and Analytic

3.3.1 Description and Priority

Description: This feature provides a user interfacing dashboard that showcases project status, analytics, and visual representation of resource scheduling and optimization results. It shall also provide real-time updates and facilitate easy navigation.

3.3.2 Stimulus/Response Sequences

- Stimulus: User logged in and proceeded to dashboard.
- System Response: " System will load the active user dashboard with relevant project information, analytics and optimization results.
- Stimulus: User acts upon the dashboard's visualization tools.
- System Response: System refreshes the visualizations based on what the user enters as input or filter.
- Stimulus: User clicks on a particular project or resource for details.
- System Response: System opens up detailed information on the selected resource or project.

3.3.3 Functional Requirements

- REQ-1: The system should provide a real-time dashboard displaying project and resource scheduling data.
- REQ-2: The system should allow interaction with the data visualizations through different plot items and charts.
- REQ-3: The system should refresh the various elements in the dashboard dynamically as changes are made by the user.

3.4 Excel Data Import and Management

3.4.1 Description and Priority

Description: This feature will allow the resource scheduling data input to be done directly by the user through Excel uploads. The system will process the file, extract the data, and show it on the dashboard, where further analysis and optimization can take place.

3.4.2 Stimulus/Response Sequences

- Stimulus: The user clicks the "Upload Excel" button.
- Sys response: The system asks him to upload an Excel file.
- Stimulus: User uploads a valid Excel file.
- Sys response: It processes it, further checks its structure, and extracts the necessary data.

- Stimulus: User uploads faulty Excel file.

- System Response: the system provides an error message with a reason regarding the uploaded file.

3.4.3 Functional Requirements

- REQ-1: The system should allow users to upload Excel files in .xlsx format.
- REQ-2: The system should check the structure of the Excel files uploaded to assure they include the appropriate columns and data type.
- REQ-3: The processing and computation of data should auto-update the dashboard for the same pertaining to scheduling and optimization.

3.5 Admin Management and Role-based Access Control

3.5.1 Description and Priority

Description: An Admin will be granted administration over user accounts, subscription plans, and system settings. This will include role-based access management: different levels of access for various types of user roles, such as Admin and User.

3.5.2 Stimulus/Response Sequences

- Stimulus: The admin logs in and accesses the admin dashboard.
- System Response: The system loads an admin interface which displays management tools and user lists.
- Stimulus: The admin updates a user's subscription or role.
- System Response: De Lorenzo's account in the system is updated to his new access level.
- Stimulus: Admin deletes a user account
- System Response The user account is deactivated and access to system features are turned off.

3.5.3 Functional Requirements

- REQ-1: The system shall allow for role-based access control where varied levels of access will be specified for the users, this may be Admin or User.
- REQ-2: The system shall have a facility to update the user account by an admin with features of updating the role and subscription.
- REQ-3: All the actions taken by admins are logged for auditing purposes.

3.6 Notifications and Alerts

3.6.1 Description and Priority

Description: The system will notify the users or provide alerts regarding vital events of subscription renewal, changes in new projects, and uploaded data.

3.6.2 Stimulus/Response Sequences

- Stimulus: A user has uploaded a new Excel file to be analyzed.
- System Response: The system provides a pop-up message upon successful data upload.
- Stimulus: A user subscription is near its expiration date
- System Response: The system notifies and asks the user to renew the subscription.

3.6.3 Functional Requirements

- REQ-1: The system shall send emails and in-app notifications about the user's important actions, including uploading and subscription expiration.

- REQ-2: The system shall grant options to users for managing notification preferences through explicit opt-in/ opt-out of specific alerts.

This system feature version for your resource scheduling and optimization platform has a great concentration on the assurance of each feature mapping with priority, stimulus/response sequences, and very detailed functional requirements.

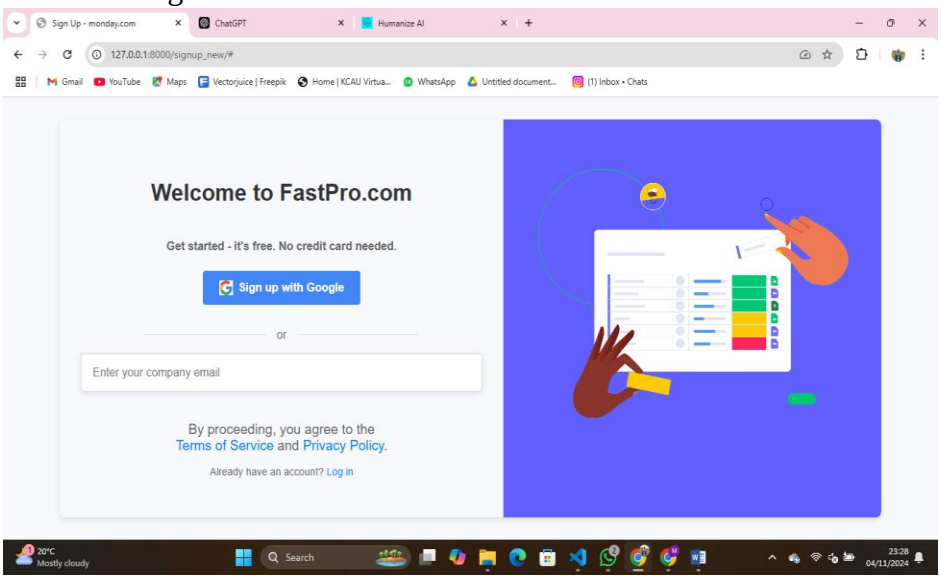
4. External Interface Requirements

4.1 User Interfaces

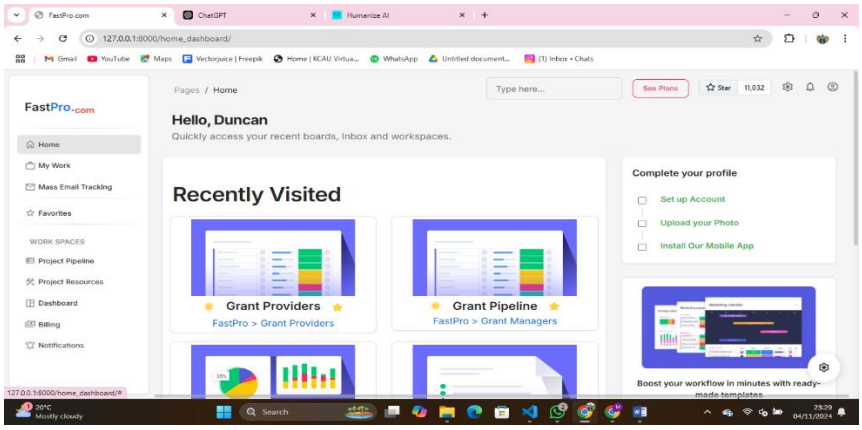
User interfaces are very important to ensure smooth interaction between the users and the system. Key characteristics for user interfaces are enlisted below.

Logical Characteristics

i. Login/Registration Screen:
- Input: Email and Password.
- Buttons: "Login", "Register", "Forgot Password".
- Functionality: Error messages in case of invalid credentials; confirmation email after successful registration.

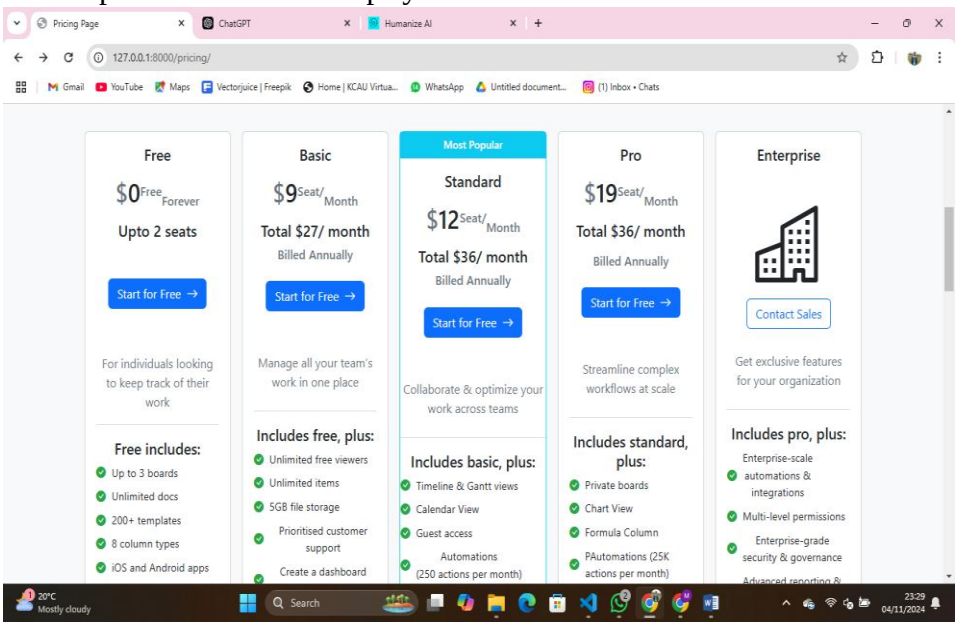


ii. Dashboard Screen:
- It should represent all the project scheduling and resource optimization details along with subscription status.
- With this, the user can interact with data visualization, like charts and tables.
- Buttons: "Projects", "View Project", "Start Optimization", "Logout".
- Navigation bar, Subscription plan status section, project information section, Resource optimization graphs.



iii. Subscription Screen

- Listing subscription plans - Free, Monthly, Yearly
- Button: "Subscribe Now", "Renew Subscription"
- Subscription and renewal expiry notices.



User Interface Guidelines

a. Screen Layout:

- The layout throughout the application should be consistent.

Responsive design- Desktop, Tablet, Mobile Device Design based on grid-based layout through Bootstrap or similar frameworks. Common Controls Standard Buttons: "Submit", "Cancel", "Back" Input fields with labels Error messages in red colour including what exactly the user should do to fix it Colours and Themes Stick to specific colour palette for readability and accessibility of Application.

- High-contrast background colours for readability.
- Help and Accessibility
- Helper icons on each screen for quick help.
- Tooltips and user instructions for accessibility

Screen Layout Constraints

- Responsiveness: The UI is fully responsive, meaning the layout needs to adapt accordingly with screen size.
- Width/Height Maximum: Buttons and input fields are among those things that must not be more than a certain width or height in order to be most usable. Good examples include the input fields, which should be 300px for bigger screens and 100% width on mobile devices.

4.2 Hardware Interfaces

The software interacts with several hardware components. The main focus will revolve around the various devices used to access the system and do the uploading of data to the system.

Device Types:

- Desktop/Laptop: Based on Windows, macOS, and Linux systems.
- Mobile Devices: Android and iOS smartphones and tablets.

Communication Protocols:

- HTTP/HTTPS for web-based communication between the client, i.e., user devices and the server.
- FTP-if needed for huge file uploads or downloads.

Hardware Interactions

The system may require specific hardware to input data, like mouse, keyboard, and possibly scanners-for barcode/QR code input.

Uploading files stored in local devices to the server for data analysis via protocols such as FTP or HTTP(S).

4.3 Software Interfaces

Interfaces to Other Software Elements:

Web Browser:

- Browsers Supported: Chrome, Firefox, Safari, and Edge.
- Communication Protocol: HTTP/HTTPS for communicating with the client and the server.

Database:

- Database Management System: SQLITE for user data storage, subscription information, project schedules, and optimization results.
- Data Flow: The application requires data regarding the schedule of the projects, results of optimization, user profiles, and subscription details for smooth working and flow, all of which are stored in the database.

Payment Gateway:

- Third-party Integration: Stripe or PayPal for subscription processing.
- Data Flow: The gateway will receive and send data regarding payment details, transaction statuses, and subscription renewal data.

Operating System:

- Server-Side OS: Ubuntu Linux or any other Server Operating System.

Libraries:

- Data Visualization: Chart.js or D3.js-for visualization of optimization data on the dashboard.
- Excel File Parsing: Pandas, openpyxl-both for backend processing in Python of Excel data.

Data Flow

Input Data:

- User credentials and project scheduling data
- User-uploaded Excel files

Output Data:

- Updates to the dashboard with processed scheduling and optimization data
- Data of payment transactions are pushed to the payment gateway.
- User Input - Error Messages and Subscription Status, etc.

4.4 Communications Interfaces

Communications Requirements

The system would provide email notifications about account verification, password recovery, subscription renewal reminders, and results of optimizations.

Protocols: SMTP for Sending Emails.

Web Browser Communication:

Communication between the user's browser and the web server using HTTP/HTTPS for all user interactions - form submission, requesting data.

File Upload:

Web UI-based uploads, such as Excel file uploads, shall use HTTP POST using multipart/form-data encoding.

Encryption

All communications shall take place via HTTPS (SSL/TLS) to guarantee data privacy for users when performing transactions, which include the log-in and payment events, including file upload.

File Transfer:

It should allow for quick uploads and data exchange within the system. The system should support a data transfer rate of 5MB/s while uploading files.

Communication Standards

Protocol: HTTP/HTTPS for web communication, SMTP for email notification, FTP/HTTP for file upload.

Security:

All communications shall be secured via SSL/TLS encryption in order to protect sensitive user data.

Two-factor authentication shall be enabled for the login to an admin dashboard as a way of locking sensitive management features.

5. Other Non-functional Requirements

5.1 Performance Requirements

Data Processing Time: The system shall process an uploaded Excel file and refresh the dashboard with data relevant to resource optimization in less than 10 seconds for files whose size is below 10 MB.

Response Time: The response time for the user interface shall be less than 2 seconds for any action taken, such as button clicks and submission of forms.

5.2 Safety Requirements

Data Loss Prevention: The system should regularly create backups of user data, including projects and subscription details, so that no data is lost.

Failure Recovery: If the failure occurs in the system - for instance, due to a crash of a server - users are not supposed to lose their unsaved data. All inputs made by users should be temporarily stored.

5.3 Security Requirements

- User Authentication: All users, especially admins, shall be authenticated by email/ password with two-factor authentication on sensitive actions such as subscription renewal or account changes.

- Data Encryption: Passwords and payment information related to sensitive user data must be encrypted before storage in the database.

5.4 Software Quality Attributes

- Maintainability: The code should be modular and well-documented, observing software design principles, for example MVC.

Usability: The user interface shall be intuitive for the users and easy to use. It shall contain instructions that are clearly presented and controls that are easy to access.

Testability: The system shall be highly testable, with automated test cases for key functionality such as login, subscription management, and data import.

6. Other Requirements

6.1 Database Requirements

Data Storage: The system shall store user information, project schedules, and subscription data in a relational database; preferably SQLite/MySQL.

- Database Structure: Tables for users, subscriptions, payments, project details, and optimization results must be accordingly normalized to avoid data redundancy.

6.2 Internationalization Requirements

- Multilingual Support: It should support multiple languages, starting with the English language, but it should be flexible enough to include more in the future.

6.3 Legal Requirements

- GDPR Compliance: The system must comply with the General Data Protection Regulation for users within the European Union.

6.4 Reuse Goals

Reusable Components: User authentication and data visualization should be done with modular components that must be reusable in other projects within the institution.

Appendix A: Glossary

API

Application Programming Interface. It is a set of protocols and development tools used to build applications and specifications through which different components of software can communicate and interact with each other.

Authentication

Authentication means confirming the identity of the user or system before it is allowed to access resources.

Authorization It is the process through which a user, or system is granted access or denied access to particular resources based on the authentication of their identity.

Back-End The server side of an application where all the business logic, data processing and storage happens.

CRUD Create, Read, Update, Delete. These four are the basic operations to be performed on data in a database or a data system.

CSV

Comma Separated Values. It is a file format that is used for storing tabular data in plain text format with every line storing a data record.

Dashboard

A user interface that presents KPIs, metrics, and other data in an easily comprehendible, usually graphical fashion for quick analysis and monitoring.

Database

A structured, organized, and typically electronically stored and accessed gathering of data, often controlled by a RDBMS like PostgreSQL or MySQL.

Encryption

The process of translating data into a code to impede unauthorized access. Encryption protects vital information, like passwords and payment information.

Excel

A Microsoft spreadsheet program and widely used for data storage, organization, and analysis in a tabular format.

FTP

File Transfer Protocol. A protocol used in a network to transfer files from a client to a server.

GDPR

General Data Protection Regulation. The Regulation under EU law concerning the protection of natural persons regarding the processing of personal data and on the free movement of such data within the European Union and the European Economic Area.

GUI

Graphical User Interface: A user interface that allows the use of all software, but instead of text-based commands, it provides facilities to view and manipulate visual elements known as buttons, icons, and windows.

HTTP/HTTPS

Hypertext Transfer Protocol / Hypertext Transfer Protocol Secure: Protocols utilized to transfer information over the web. HTTPS makes sure that the communication is encrypted.

JSON

JavaScript Object Notation. A lightweight data-interchange format that is easy for humans to read and write and easy for machines to parse and generate.

KPI

Key Performance Indicator. A measurable value that demonstrates how effectively a system or organisation is achieving its key objectives.

MVP

Minimum Viable Product. The initial version of a product that has just enough features to satisfy early adopters and elicit feedback for future development.

MVC

Model-View-Controller. A software architecture pattern that separates an application into three interconnected components: the model, the view, and the controller.

OAuth

Open Authorization. The standard protocol that is used for granting access to a user's information across third-party services without exposing their password.

Optimization

The process of making a system as effective or functional as possible with generally involving resource scheduling, cost reduction, or performance enhancement.

Pandas

A Python library for manipulation and analysis of data. It is very effective when dealing with structured data such as CSVs and Excel.

PostgreSQL

An open-source RDBMS that is robust, scalable, and supports most advanced features including transactions and stored procedures.

SaaS

Software as a Service. A software distribution model in which applications are hosted by a third-party provider and made available to customers over the internet.

SRS

Software Requirements Specification: A document that describes the software product to be developed including the functional and non-functional requirements, interfaces, and constraints.

SSL/TLS

Secure Sockets Layer / Transport Layer Security: Cryptographic protocols designed to provide secure communication over a computer network. Main use is for securing HTTP (web) traffic.

SMTP

Simple Mail Transfer Protocol: Protocol for sending email messages between servers.

Subscription

A pricing model in which customers continually pay to access something but usually in multi-tiered levels - e.g., Free, Monthly, and Yearly. Two-factor Authentication (2FA) A security process whereby someone has to use two different methods of proving they are who they say they are to get into a system. This often consists of a password and an authentication code sent to their mobile device. UI

User Interface. It is an area where the human and machine meet. The UI comprises everything that allows the user to communicate input and receive information with the system.

UX

User Experience. The overall feeling the user has regarding his or her interaction with a system based on efficiency, ease of use, and level of satisfaction.

XML

Extensible Markup Language. Human- and machine-readable markup language for storing and transporting data.