

Requirement Specification: Complex Task System

Change the World

Readers guide.....	3
Introduction.....	4
Background.....	4
Purpose & goals.....	4
Scope.....	4
Functional requirements.....	5
User Authentication System.....	6
Project Management System.....	7
Project Modelling System.....	10
Realtime Collaboration System.....	12
Payment System.....	14
Account Management System.....	17
Non-functional requirements.....	18
User interface design.....	19
Signup / signin.....	19
Profile.....	20
Projects.....	21
Project.....	22
Support - just a top-down list of FAQs, for now? scope should be refined.....	22
Architecture.....	23
Deployment.....	23
Appendix.....	24
Changelog.....	24

Readers guide

The purpose of this requirement specification, is to effectively communicate the business goals, as well as the elicited requirements for the system. This is about defining the vision for the system. Initially, the functional requirements of the system are presented, broken down by themes, epics and ultimately user stories. Afterwards, broader and softer requirements of the system are noted as non-functional requirements. Lastly user interface design requirements are listed with wireframes to illustrate how the user stories should be implemented.

There is also a comments section. This is for educational purposes, as it is a place to list improvements and many reflections deemed relevant for the quality of the requirement specification and or pre-analysis.

Introduction

Background

In today's fast-paced environment, managing complex tasks can be daunting, often requiring significant resources that are already scarce. There is a critical need to streamline these processes to enhance efficiency and reduce the resource burden to complete projects and achieve tangible results. But the market hasn't been able to address this issue, as there is a gap between visualizing complex tasks and providing exhaustingly detailed information.

Purpose & goals

Specify and develop a system that enables both individuals and organizations to isolate the dimensions of a complex task in a visually compelling way and allowing for detailed input

- **Reduce resources spent on complex tasks**, by developing a system that addresses issues in existing processes and effectively combine these into an accessible product, measured by rating based user feedback using a scale from 1 to 10 (1 being least unsatisfied, 10 being most satisfied).
- **Appeal to multiple target groups**, to improve the sustainability of the system, which should successfully be able to generate value for all users which need to complete a complex task. Initially the target is to reach two target groups: students and project managers, which is evaluated based on their feedback → was the system accepted or not? - rating above 7 out of 10
- **Create a profitable revenue stream**, to ensure sustainability for the business and allowing for future capital investing in the business, measured by monthly revenue and costs.
- **Facilitating a low-maintenance operational system**, to reduce personal time needed for managing the system by ensuring robustness is a value proposition for myself, as well as users of the system, measured by weekly time spent addressing issues in the system. Initial threshold is 7 hours spent weekly on system maintenance as well as customer support.

Scope

A system enabling users to isolate dimensions of complex tasks, using visuals while allowing for exhaustingly detailed input. This includes the following:

- User Authentication System
- Project Management System
- Project Modelling System
- Realtime Collaboration System
- Payment System
- Account Management System

Functional requirements

The functional requirements of the system are broken down into its sub-systems, which is interpreted as *themes* from the Agile Methodology. Thereafter, the themes are broken down by *epics* and lastly these are broken down by their atomic *user stories*, which are the tangible requirements for the overall system.

- User Authentication System
 - ◆ User registration
 - ◆ Password management
- Project Management System
 - ◆ Resource Management
 - ◆ User Management
 - ◆ Exporting the project
- Project Modelling System
 - ◆ Project Visualizing
 - ◆ Detailed Content Management
- Realtime Collaboration System
 - ◆ Nodes
 - ◆ Models
 - ◆ User Activity
- Payment System
 - ◆ Pricing tier
 - ◆ Billing
 - ◆ Invoicing
- Account Management System
 - ◆ Profile
 - ◆ Payment
- Organizational Capabilities (?)
 - ◆ ***Use cases related to this haven't been specified, as the business hasn't yet decided on their value proposition in regards to organizational services.**
 - ◆ **The initial project is targeting students and solo professionals, which doesn't require organizational capabilities.**

User Authentication System

User Registration, Password Management

User Registration

- **User story:** As a user, I want to sign up to the service, so I can get started using it.
 - **Description:** Allow the user to sign up.
 - **Acceptance criteria**
 - Valid credentials are provided
 - Email, password
 - Password is at least six characters
 - Third party providers (tbd.)
 - User is notified if credentials are invalid
-
- **User story:** As a user, I want to sign in to the service, so I can continue where I left.
 - **Description:** Allow the user to sign in.
 - **Acceptance criteria**
 - Valid credentials are provided
 - Logging in gains the user access to their account and associated resources
 - User is notified if credentials are invalid
-

Password Management

- **User story:** As a user, I want to reset my password in case I've forgotten it.
 - **Description:** Allow the user to reset their password.
 - **Acceptance criteria**
 - The user provides a valid email
 - The email is registered in the system
 - The user receives a reset link in their email
 - The user can provide a new password
 - The new password is at least six characters

Project Management System

Resource Management, User Management

Resource Management

- **User story:** As a user, I want to create a new project, so I can get started with my complex task.
 - **Description:** Allow the user to create a new project.
 - **Acceptance criteria**
 - The user is logged in
 - The user provides a valid name for the project
 - The user has not exceeded their limit, depending on their subscription.
-
- **User story:** As a user, I want to view all my projects, so I can get an overview of what I have going on.
 - **Description:** Allow the user to view all their projects.
 - **Acceptance criteria**
 - The user is logged in
 - A list of the user's projects is displayed
 - Project information is shown, such as:
 - Name
 - Users (avatar or something)
 - Preview of project (glhf)
-
- **User story:** As a user, I want to view all projects I've been invited to, so I can get an overview of what I have going on.
 - **Description:** Allow the user to view all projects they've been invited to.
 - **Acceptance criteria**
 - The user is logged in
 - A list of projects the user has been invited to is displayed
 - Project information is shown, such as:
 - Name
 - Users (use avatar for display)
 - Preview of project (image snapshot of the model, glhf)
-
- **User story:** As a user, I want to access a project, so I can get started working on my complex task.
 - **Description:** Allow the user to navigate to a specified project.
 - **Acceptance criteria**
 - The user is logged in
 - The user can navigate to the specified project
 - The project's data is displayed in the model editor
-

- **User story:** As a user, I want to rename a project, so I can ensure the name reflects its context.
 - **Description:** Allow the user to rename a project.
 - **Acceptance criteria**
 - The user is logged in
 - The user can provide a valid name for the project
 - At least five characters
 - The project's name is changed
 - The project belongs to the user
-
- **User story:** As a user, I want to delete a project, so I can limit resources used and also just clean up unneeded projects.
 - **Description:** Allow the user to delete a project.
 - **Acceptance criteria**
 - The user is logged in
 - The project belongs to the user
 - User has went through a mis-click safeguard
-

User Management

- **User story:** As a user, I want to invite other users to a project, so they can access my project and contribute.
 - **Description:** Allow the user to invite other users to a project.
 - **Acceptance criteria**
 - The user is logged in
 - The user is the owner of the project
 - The user provides a valid email
 - Email doesn't have to be registered in the system, it may be a soon-to-be user, who has to sign up first.
 - The user is using a pricing tier that enables collaboration features
 - And hasn't exceeded their project collaborators limit (standard subscription)
-
- **User story:** As a user, I want to view all users that have access to a project, so I can get an overview of the 'team'.
 - **Description:** Allow the user to view all users associated with the project.
 - **Acceptance criteria**
 - The user is logged in
 - The user has access to the project
 - Owner or invited
 - The users of the project is displayed
-

- **User story:** As a user, I want to remove a user from the project, so I can ensure integrity of the project's content.
 - **Description:** Allow the user to remove a user from the project.
 - **Acceptance criteria**
 - The user is logged in
 - The user is the owner of the project
 - The removed user receives an notification
 - Email
 - In-app(?)
-

Exporting the Project

- **User story:** As a user, I want to export the project to a .docx file, so I can present it for relevant stakeholders.
 - **Description:** Allow the user to export the project to a .docx file.
 - **Acceptance criteria**
 - The user is logged in
 - The user has a subscription that allows exporting projects
 - The user hasn't exceeded the limit no. of monthly exports
-
- **User story:** As a user, I want to specify nodes to omit when exporting the project, so I can get only what I need exported.
 - **Description:** Allow the user to omit nodes when exporting.
 - **Acceptance criteria**
 - The user is logged in
 - The user can choose whether a node is omitted
-
- **User story:** As a user, I want to specify the order when exporting my project, so I can ensure it is correctly structured as per my needs.
 - **Description:** Allow the user to order nodes to be exported.
 - **Acceptance criteria**
 - The user is logged in
 - The user can specify an order for nodes to be exported
-
- **User story:** As a user, I want to view the current export order of the project, so I can get a grasp of the current project as is.
 - **Description:** Allow the user to view the current export order.
 - **Acceptance criteria**
 - The user is logged in
 - The user is displayed a list of nodes corresponding to their order number

Project Modelling System

Resource Management, User Management

Project Visualizing

- **User story:** As a user, I want to make diagrams / models, so I can isolate the components of my complex task.
 - **Description:** Allow the user to make diagrams.
 - **Acceptance criteria**
 - The user is logged in
 - The user can seamlessly make diagrams, with a palette of more diagrams than they may need to use, ensuring coverability and relevancy
 - The diagram can be stored and accessed later
 - The diagram is capable of programmatic manipulation
 - **User story:** As a user, I want to have multiple models for a single project, so I can be creative and truly isolate dimensions of my complex task in a compelling way.
 - **Description:** Allow the user to create multiple tabs, each one with a canvas to make diagrams.
 - **Acceptance criteria**
 - The user is logged in
 - The project doesn't exceed any subscription limitations
 - **User story:** As a user, I want to rename a model, so I can ensure they reflect my needs for the project.
 - **Description:** Allow the user to rename a model.
 - **Acceptance criteria**
 - The user is logged in
 - The user provides a valid name for the model
 - **User story:** As a user, I want to rearrange models in the project, so I can change up as I deem necessary for maintaining the project.
 - **Description:** Allow the user to rearrange the order of the models.
 - **Acceptance criteria**
 - The user logged in
 - **User story:** As a user, I want to delete a model, so I can get rid of unneeded complexity in the project.
 - **Description:** Allow the user to delete a model in the project.
 - **Acceptance criteria**
 - The user is logged in
 - The user has gone through a mis-click safeguard
-

Detailed Content Management

- **User story:** As a user, I want to provide detailed content to a part of the project's model, so I can flesh out and work towards completing the complex task.
 - **Description:** Allow the user to provide content to a model node in the project.
 - **Acceptance criteria**
 - The user is logged in
 - The content is one or a combination of the following type of media:
 - Text
 - Tables
 - Images
-
- **User story:** As a user, I want to make comments, so I can revisit and adjust my work later or properly communicate suggestions to others' work.
 - **Description:** Allow the user to create comments.
 - **Acceptance criteria**
 - The user's name / avatar is displayed in the comment
 - The user can provide text in the comment
-
- **User story:** As a user, I want to view comments for the project, so I can get a grasp on immediate issues.
 - **Description:** Allow the user to view comments for the project.
 - **Acceptance criteria**
 - The user can view a list of all comments
 - Each comment has relevant metadata, including:
 - Author
 - Timestamp
 - Point in the project the comment is bound to (model, text editor...)
-
- **User story:** As a user, I want to delete a comment, so I can clean up unneeded things in my project.
 - **Description:** Allow the user to delete a comment.
 - **Acceptance criteria**
 - The user can specify a comment to delete
 - The comment is deleted when a user has been through a mis-click safeguard
-
- **User story:** As a user, I want to view the current no. of characters in the project (text-wise), so I can access whether I hit a length limitation.
 - **Description:** Allow the user to view total no. of text characters in the project.
 - **Acceptance criteria**
 - All nodes are included, except for non-exported nodes
 - The count always reflects the current state of the project
 - There should be displayed w. and without whitespaces.
-
- **User story:** As a user, I want to resize an image in the details editor, so I can properly view things.

- **Description:** Allow the user to resize an image in the details editor.
 - **Acceptance criteria**
 - The user can resize an image in the editor
 - The image does not exceed the editor's bounds
 - The image can't be made too small, so it is borderline invisible
-

- **User story:** As a user, I want to position an image within the details editor, so I can properly view things.
 - **Description:** Allow the user to position an image in the details editor.
 - **Acceptance criteria**
 - The user can position an image in the editor
 - The image is contained within the editor's bounds
-

Realtime Collaboration System

Nodes, Models, User Activity

Nodes

- **User story:** As a user, I want to see where the user is writing inside a node's editor, if I am in the same node, so I can properly collaborate.
 - **Description:** Allow a user to see other users' writing marker inside a node, if they are at the same node.
 - **Acceptance criteria**
 - Each user has a unique color
 - The marker is visible, if they are inside the same node
-

Models

- **User story:** As a user, I want to see other users' cursor inside a model, if I am at the same model, so I can follow other's work.
 - **Description:** Allow the user to view other users' cursor inside a model, if they are in the same model.
 - **Acceptance criteria**
 - Each user has a unique color
 - The cursor is visible, if they are inside the same model
 - **User story:** As a user, I want to see when other users have selected a model element, so I can follow their activity.
 - **Description:** Allow the user to view when other users have selected a model element.
 - **Acceptance criteria**
 - Each user has a unique color
 - The element should be outlined with the user's color
-

User Activity

-
- **User story:** As a user, I want to see all connected users in the project, so I can monitor the activity in the project, while i'm working.
 - **Description:** Allow the user to view all connected users in the project.
 - **Acceptance criteria**
 - Each user has a unique color
 - Only connected users are shown
 - Each user has their avatar displayed
-
- **User story:** As a user, I want to be notified whenever the project has been exported, so I can access information integrity of the project.
 - **Description:** Allow the user to be notified whenever the project is exported.
 - **Acceptance criteria**
 - The notification is displayed in a non-invasive way
 - The notification is displayed regardless of location the user is currently in the project
 - The notification includes at least the following data:
 - The name and or email of the user who exported the project
 - Timestamp of export
-
- **User story:** As a user, I want to preview the export of the project, so I can get a grip on what the end product looks like.
 - **Description:** Allow the user to preview the export of the project.
 - **Acceptance criteria**
 - The preview is non-editable
 - The preview resembles a .docx / pdf file format
-

Payment System

Subscription, Payment Details, Billing, Invoicing

Subscription

- **User story:** As a user, I want to view all possible subscription tiers and their features, so I can make a decision on which tier fits my needs.
 - **Description:** Allow the user to view all subscription tiers.
 - **Acceptance criteria**
 - All subscription tiers are displayed
 - Features of the tier's are displayed
 - The tiers are displayed in a comparable format
 - **User story:** As a user, I want to select a subscription tier to use, so I can get started using its features.
 - **Description:** Allow the user to select a subscription tier.
 - **Acceptance criteria**
 - Valid payment credentials
 - The user has already provided valid payment credentials in their profile
 - Or, the user provides new credentials
 - Making the flow / conversion to paid user seamless
-

Payment Details

- **User story:** As a user, I want to provide payment details, so I can pay for the service with the correct payment method.
 - **Description:** Allow the user to provide payment details.
 - **Acceptance criteria**
 - The user provides valid payment details
 - The payment details is not stored directly in the system
 - The user can at most provide one set of payment details
-

- **User story:** As a user, I want to view my payment details, so I can ensure I am using the correct payment method.
 - **Description:** Allow the user to view their payment details.
 - **Acceptance criteria**
 - The user can view their payment details

- **User story:** As a user, I want to edit my payment details, if I need to update it for some reason.
 - **Description:** Allow the user to edit their payment details.
 - **Acceptance criteria**
 - Valid payment details are provided

- **User story:** As a user, I want to remove my payment details, if I for some reason don't want them present.
 - **Description:** Allow the user to delete their payment details.
 - **Acceptance criteria**
 - The payment details are completely forgotten in every related system
 - The user is notified, if they are subscribed to a plan
 - The user completes a mis-click safeguard

Billing

- **User story:** As a user, I want to be billed automatically every period, so I can continue using the service without interruption.
 - **Description:** Allow the user to be billed automatically.
 - **Acceptance criteria**
 - The user's specified payment details are used
 - The user is billed every month, from their start date

- **User story:** As a user, I want to be notified if the payment details I've provided were rejected, so I can address the issue and avoid interruptions.
 - **Description:** Allow the user to be notified if their payment details were rejected.
 - **Acceptance criteria**
 - The user is notified by their registered email
 - The user's account is limited to free features until they've paid
 - Data is removed after a week if they haven't paid
 - An email is sent 5 days after the rejection, if they still haven't paid

Invoicing

- **User story:** As a user, I want to receive an invoice whenever I am billed for my subscription, so I can keep up with costs.
 - **Description:** Allow the user to receive an invoice whenever they are billed.
 - **Acceptance criteria**
 - The user receives an invoice by their registered email once billed
 - The invoice is stored somewhere internally
 - The invoice is sent as a .pdf document
-
- **User story:** As a user, I want to view all my invoices, so I can get a grip on total costs.
 - **Description:** Allow the user to view all their invoices.
 - **Acceptance criteria**
 - A listing of invoices is displayed
 - The user can filter their invoices
 - Date (oldest, newest)
-

Account Management System

Profile

Profile

- **User story:** As a user, I want to view my profile details, so I can access whether they are correct.
 - **Description:** Allow the user to view their profile details.
 - **Acceptance criteria**
 - Email is displayed
 - Avatar is displayed
 - Image or stock avatar
-
- **User story:** As a user, I want to edit my email, so I can ensure it is up to date.
 - **Description:** Allow the user to edit their email.
 - **Acceptance criteria**
 - The user provides a valid email
 - A verification link is sent to the email, before changing
-
- **User story:** As a user, I want to edit my avatar, so I can decide how to present myself.
 - **Description:** Allow the user to change their avatar.
 - **Acceptance criteria**
 - The user can upload an image
 - The user can remove the current image, defaulting to a stock avatar
-

Non-functional requirements

User Experience

It is important the system is easy to use and provides a memorable user experience. This is achieved by the following:

- Few clicks → When a user needs to do something, it should be accessible in less than 3 clicks.
- No deeply nested pages → There should be no pages only accessible from other pages. Seek to use modals as a possible replacement instead.
- Consistent layout → User's must be able to easily recognize the layout on every page.

Payment

The system should avoid having to comply with PCI compliance requirements, which primarily means it should avoid storing payment details.

Security

The system must have implemented common security initiatives, this includes SSL and token management for communicating between internal services.

Furthermore, users' passwords must be hashed before being stored in a database.

Speed

The speed of the system must not affect the user experience negatively. This has different criteria for aspects of the system, but in general a user should be able to continue with their work within 400ms of taking an action.

Theme

The interface must adhere to a standardized theme color palette.

User interface design

How the user stories must be implemented is first broken down by the fitting no. of pages. The elicited pages for the system are:

- Signup
- Signin
- Profile
- Projects
- Project
- Support - just a top-down list of FAQs, for now? scope should be refined


Signup / signin

signup / login

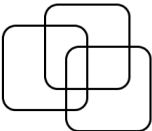
<Email>

<Password>

<LoginButton>




Profile




Projects

Profile

Support






dummy@dummy.com



Invoices

Sub	Amount	Date



Payment Details

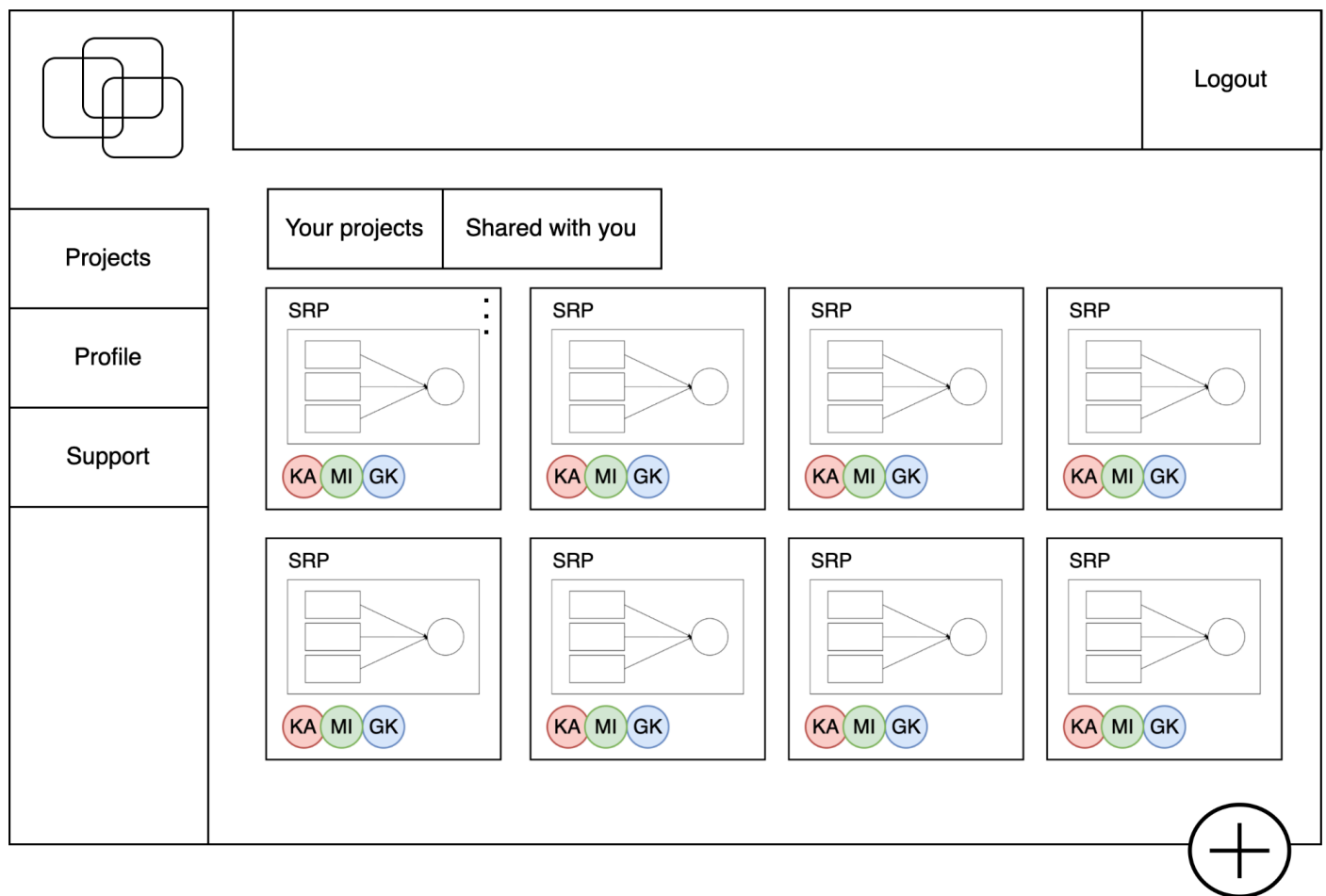
have a table of the different subscription tables, and a button named "Change subscription"

-

Button name depends on if the user already is subscribed to something, if not, its named "Subscribe"

And a modal, that ensures the user wants to change etc.

Projects



Project

Projects

Profile

Support

Model 1 / ✕

Model 2 / ✕

Model 3 / ✕

⊕

<toolbar>

Logout

Users

KA MI GK ⊕

Selected node name

<toolbar>

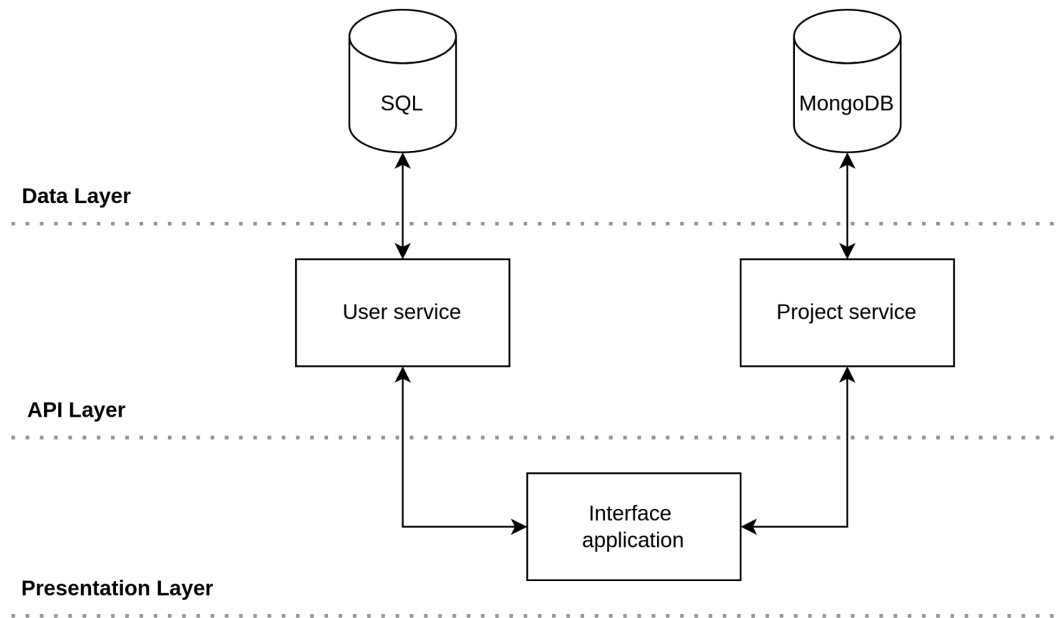
Write here...

Support - just a top-down list of FAQs, for now? scope should be refined

Architecture

The system could be built using microservices, but this is a suggestion, as the individual responsibilities for both services are limited, and can seamlessly be combined into a monolith and separated later. This would keep costs significantly lower.

If it is deemed proper to build the system using microservices, this model below demonstrates the conceptual architecture.



Where the Project Service is responsible for all logic regarding a 'project' and how users interact with them. The User Service is responsible for all logic regarding user's profile and billing information and authentication.

There are important considerations for the choice of database(s). Data specified for a project is subject to frequent changes and constitutes that a database that can adapt to a volatile nature should be used, which is why MongoDB is suggested to support the business's need for iteration and implementation of new features, requiring adjustments in the data model.

Furthermore, SQL is suggested as it enforces several security constraints by its nature. The type of SQL implementation is not specified and may be either MySQL, PostgreSQL etc.

Deployment

The system is initially to be hosted on Google Cloud, as it is possible to achieve low costs for both hosted applications as well as databases, while also enabling great integrations with a multitude of relevant solutions (Google Workspace, Cloud SQL, Firebase).

Furthermore, there must not be implemented environments for production, stage, development as this intensively drives up the costs (capital & time) and slightly hinders the primary goal of the business: to develop a MVP and gather feedback, while being able to implement and ship new features quickly. For the deployment process, github and docker should be used, to ensure consistent deployment and control over the environment.

Appendix

Changelog