

Requirement Specification: Portal

Change the world

Readers guide.....	3
Introduction.....	4
Background.....	4
Purpose & goals.....	4
Description of users.....	4
Functional requirements.....	5
User registration and authentication system.....	5
User management.....	7
Billing management (POSTPONED).....	14
Case management.....	15
Application integration capabilities.....	16
Customer support features (POSTPONED).....	17
Non-functional requirements.....	18
Data requirements.....	19
Appendix.....	20
Changelog.....	20

Readers guide

This requirement specification document structure starts with the context of the project, including the strategic situation and how the project is aimed to improve it.

The requirements themselves, are formulated as user stories which is the inquired functionality of the system. The user stories may be supported by BPMN diagrams, as there may be specific underlying requirements to the process. Validation criterias are included for every user story. All this ensures that the users needs are met and the underlying process reflects the organization's needs.

Introduction

Background

Altiore is a newly started company that aims to provide solutions with focus on 'working better with risk assessments'. These solutions will be separate applications which need to be easily accessible for the customer.

Furthermore, the user should be able to manage their information in a single place. Information is vaguely defined, but will most likely include billing and user management.

This solution will be the main entrypoint for customers, with regards to both signup and signin.

Purpose & goals

Enable Altiore to distribute their future offerings effectively through a cloud interface.

- **Create a foundation for future offerings** so new solutions can easily be integrated into the existing architecture, measured by time required to integrate a new solution and the number of needed modifications.
- **Develop a main entrypoint for users** so they can be converted into customers, measured by the success rate of user sign-ups and sign-ins.
- **Centralize offerings** enabling customers to seamlessly access solutions in one place.
- **Define and implement pricing model** to specify revenue streams, ensure profitability and allow users to pay for used services, measured by diversity of pricing models adopted.
- **Specify customer support scope** so an adequate level of support is available depending on the customer and business use case. This may (also) result in a FAQ page. The pricing model should also tackle this. Measured by a cost benefit of resources needed to support a customer.

Description of users

Managers are the high level users that need to manage other users within the organization and be the one responsible for handling billing information.

Employees are the operational users, responsible for performing the 'grunt' work within the solution. They have limited access to organization and billing management.

Administrators are Altiore's own support employees, responsible for addressing customer issues and resolving problems. This means they should have access to all data in the solution.

Developers are Altiore's own employees, who develop and integrate new solutions into the portal. They don't need access to the customers data, but require integration capabilities.

Functional requirements

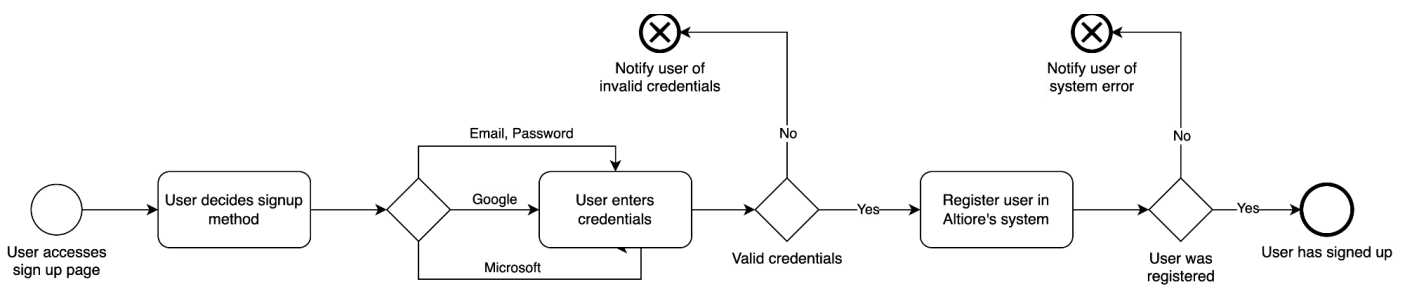
It is important to note that managers and employees do not strictly differ in this system, as the manager of a group may elevate an employee to the level as the manager. These user stories are written in the 'common' table, where solely manager or employee user stories are put in their respective table below.

User registration and authentication system

User registration, Password management, User validation

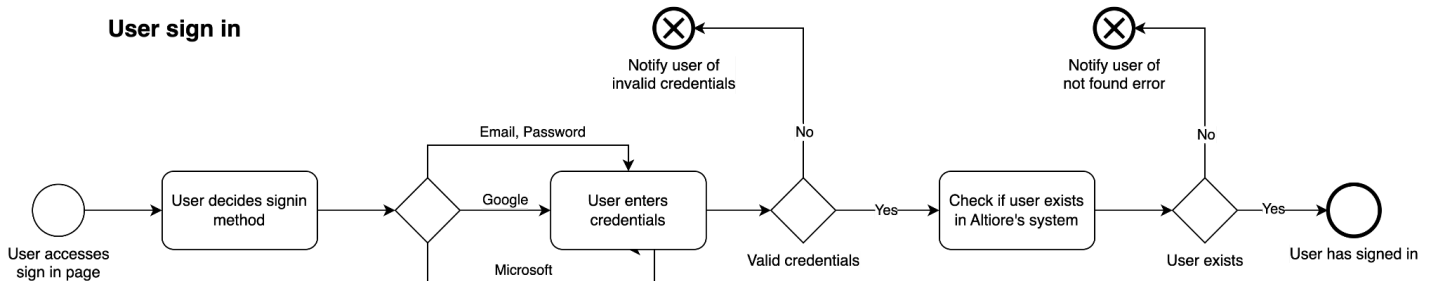
User registration

- **User story:** As a new user, I want to sign up, so I can use the portal.
 - **Description:** Allow new users to sign up.
 - **Validation criteria**
 - The user can sign up using email and a password.
 - The system sends a verification email.
 - The user can verify and complete sign up.
 - The password is at least six characters.
 - The user can sign up using a Google account.
 - Optional requirement for now, Firebase changes bricked this..
 - The user can sign up using a Microsoft account.
 - Optional requirement for now, Firebase changes bricked this..



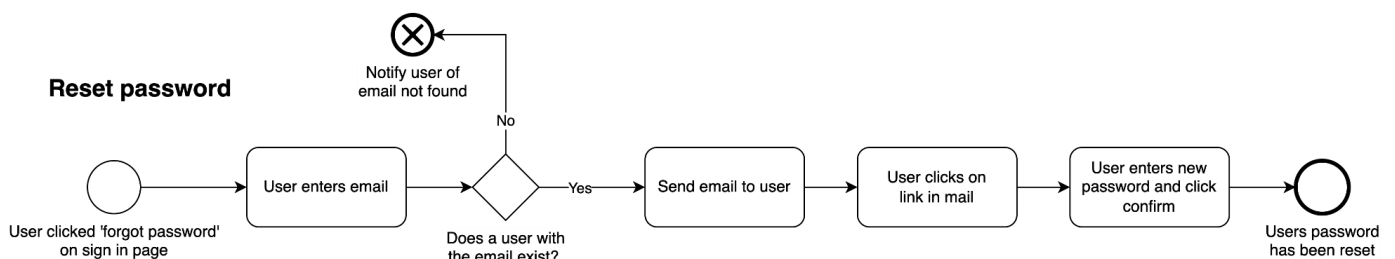
- **User story:** As a user, I want to sign in, so I can use the portal with my existing account.
 - **Description:** Allow users to sign in.
 - **Validation criteria**
 - The user can sign in using an email and a password.
 - The user is informed in case their credentials are invalid.

- The user can sign in using a Google account.
 - Optional requirement for now, Firebase changes bricked this..
- The user can sign in using a Microsoft account.
 - Optional requirement for now, Firebase changes bricked this..



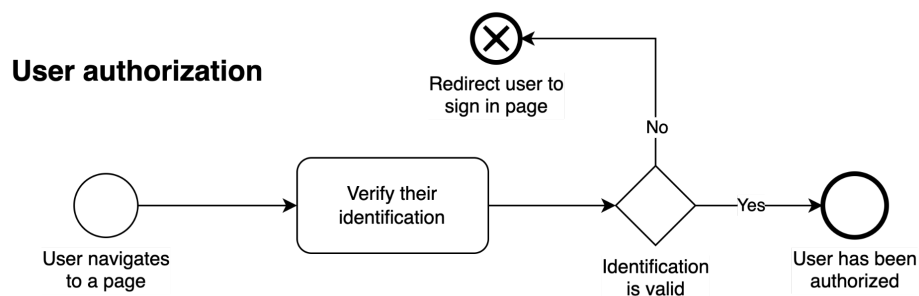
Password management

- **User story:** As a user, I want to reset my password if I forget it, so I can keep using my existing account.
 - **Description:** Allow a user to reset their password.
 - **Validation criteria**
 - The user can invoke a 'forgot password' process.
 - The user receives an email, allowing them to reset their password.
 - The user can provide a new password
 - The new password is at least six characters.
 - The user passes a safeguard, ensuring they provide the intended password.
 - The user is navigated back to login when they have successfully reset their password.



User validation

- **User story:** As an administrator, I want to ensure only authorized users have access to the portal, so other users' data isn't compromised and I can minimize support requests.
 - **Description:** Authorize users continually.
 - **Validation criteria**
 - An attempt to authorize the user is done throughout the portal.
 - Unauthorized users are returned to sign in.

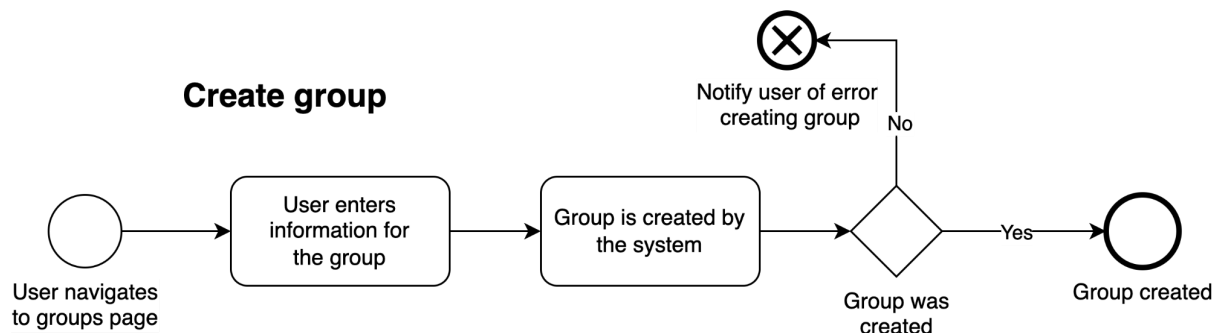


User management

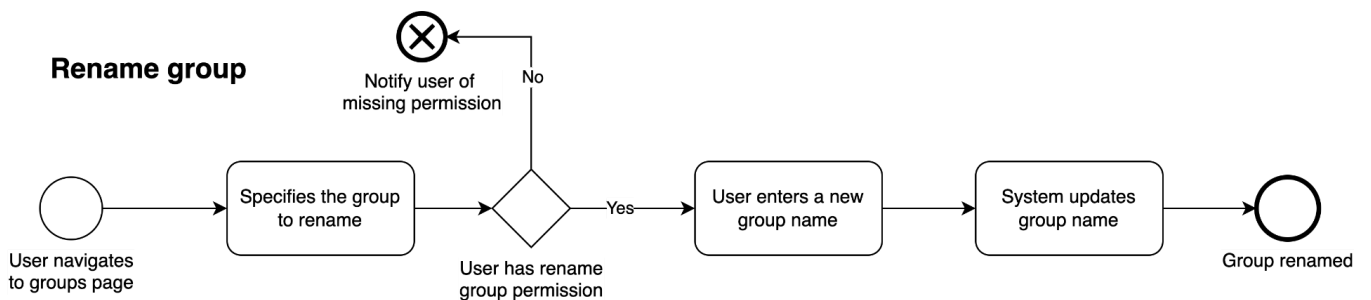
Groups, Users, Roles & Permissions, Logging

Groups

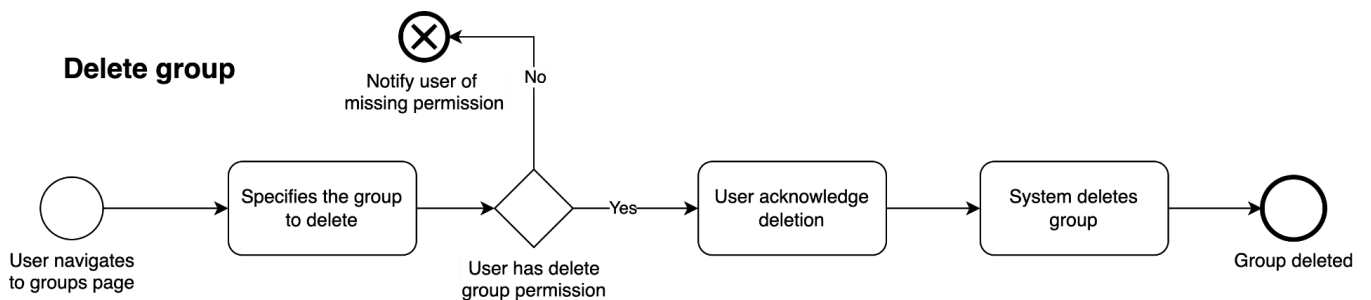
- **User story:** As a user, I want to create a new group, so I can separate my work.
 - **Description:** Allow the user to create new groups.
 - **Validation criteria**
 - The user can provide a name for the new group.
 - The user can view the new group.
 - There is an indication, if an error occurred.



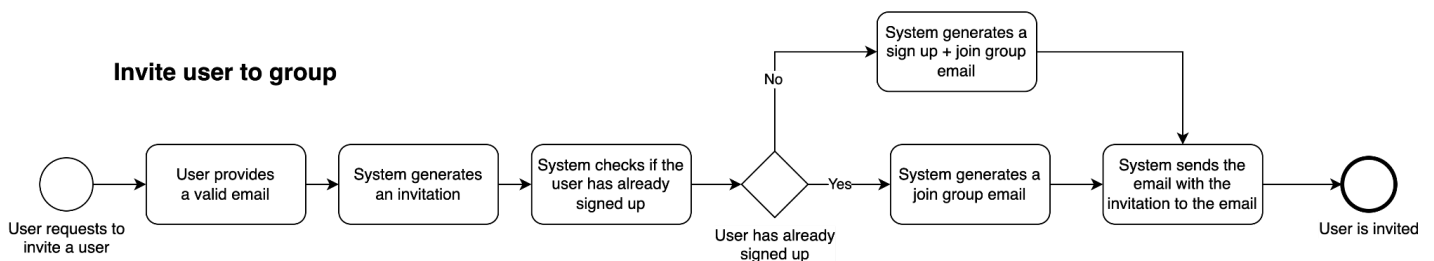
- **User story:** As a user, I want to rename a group if I feel the need to.
 - **Description:** Allow the user to rename a group, if they have the permission to do so.
 - **Validation criteria**
 - The user can specify the new name for the group.
 - The user has the permission to rename the group.



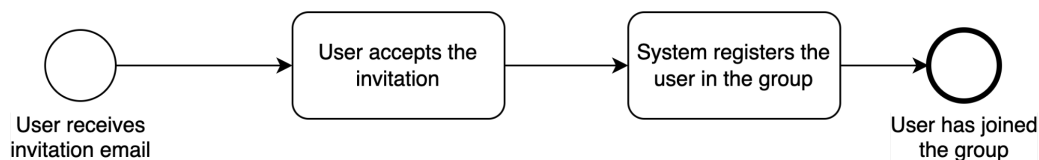
- **User story:** As a user, I want to delete a group, so I can remove unneeded groups.
 - **Description:** Allow users to delete a group.
 - **Validation criteria**
 - The user can delete a specific group.
 - The user has the permission to delete the group.



- **User story:** As a user, I want to invite other users to the group, so I can work together with others.
 - **Description:** Allow the user of a group to invite other users.
 - **Validation criteria**
 - The user can specify which user to invite by email.
 - If the user exists, they will receive an email notification.
 - If the user doesn't exist, they will receive an email, instructing them to sign up.
 - Has the permission to invite users.



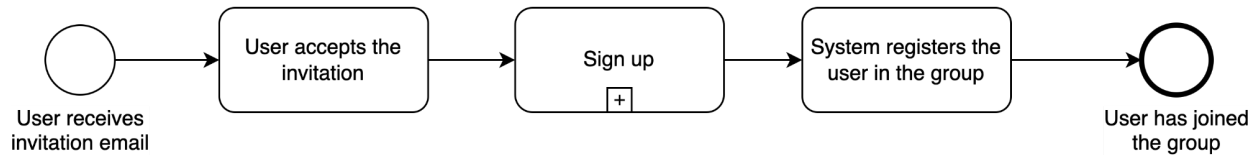
-
- **User story:** As an existing user, I want to join a group using an invitation I've received, so I can team up and collaborate.
 - **Description:** Allow the user to join a group, using an invitation.
 - **Validation criteria**
 - The user has an invitation to join.
 - The user can accept the invitation.
 - The user can see the name of the group they've been invited to.



-
- **User story:** As a non-existent user, I want to sign up and join a group using an invitation I've received, so I can team up and collaborate.
 - **Description:** Allow a non-existent user to sign up and join a group using an invitation.

- **Validation criteria**

- The user has an invitation to join.
- The user is instructed on how to sign up.
- The user provides valid credentials.
- On successful sign up, the user is associated with the group.

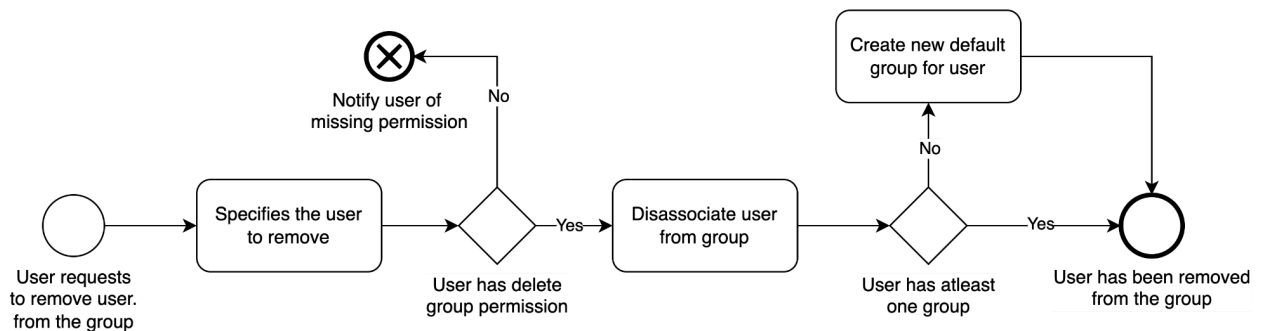


- **User story:** As a user, I want to remove other users from the group, so I can ensure only relevant users have access to data belonging to the group.

- **Description:** Allow the user of a group to remove other users.

- **Validation criteria**

- The user can specify which user to remove.
- The removed user receives an email notification.
- Has permission to remove users from the group.



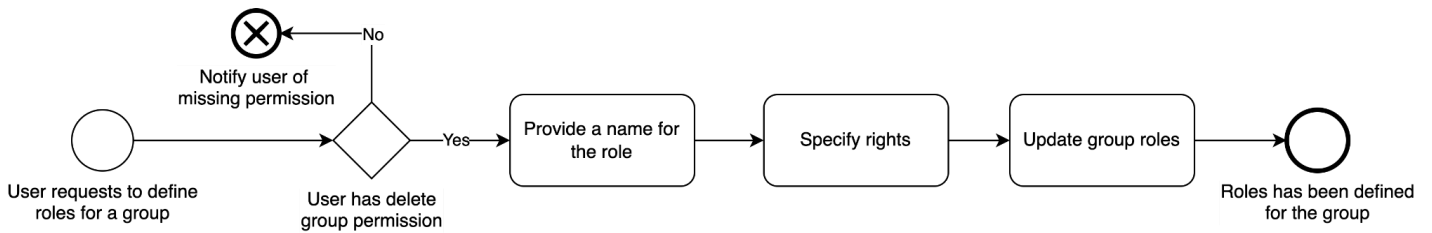
Roles & permissions

- **User story:** As a manager, I want to define roles, so I can quickly give a user within the group certain rights.

- **Description:** Allow the manager to define roles.

- **Validation criteria**

- The user can provide a name for the role.
- The user can specify rights for the role. (rights should be listed here?)
- The user has permission to define roles.



- **User story:** As a manager, I want to view a group's roles, so I can get an overview.

- **Description:** Allow the manager to view the roles within the group.

- **Validation criteria**

- The user can view the group's roles.

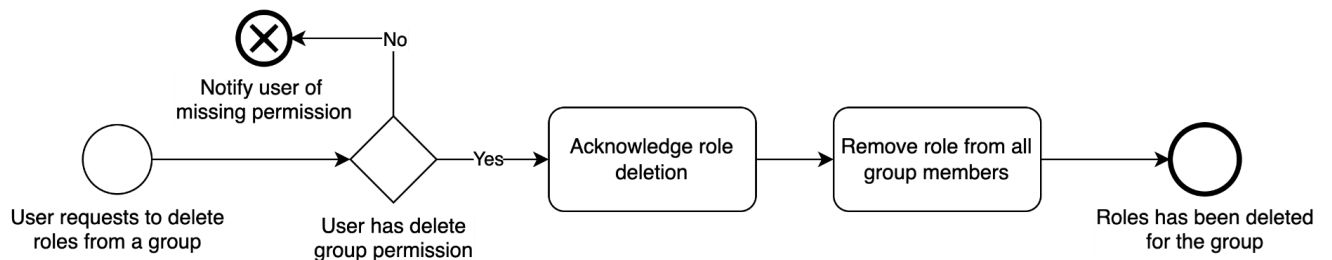
- **User story:** As a manager, I want to delete roles, so I can remove unnecessary roles.

- **Description:** Allow the manager to delete roles

- **Validation criteria**

- The user can delete roles.

- The user has permission to delete roles.



- **User story:** As a user, I want to give a user within the group one or more roles, so I can elevate them.

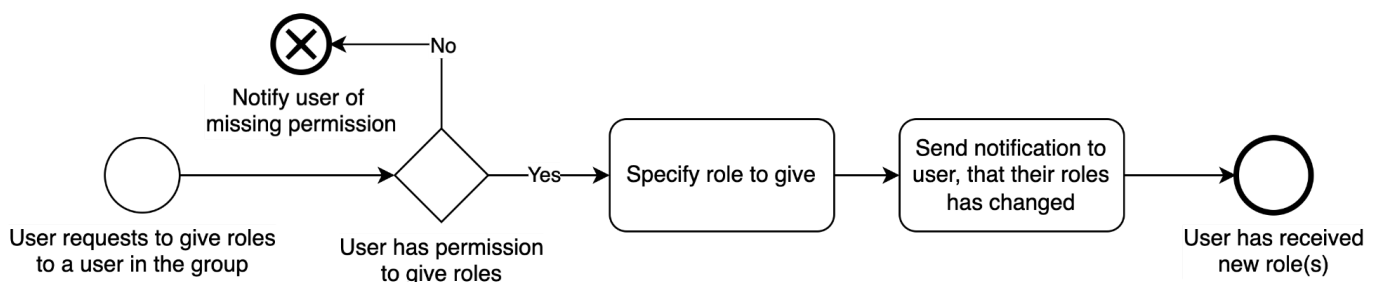
- **Description:** Allow the user to set one or more roles.

- **Validation criteria**

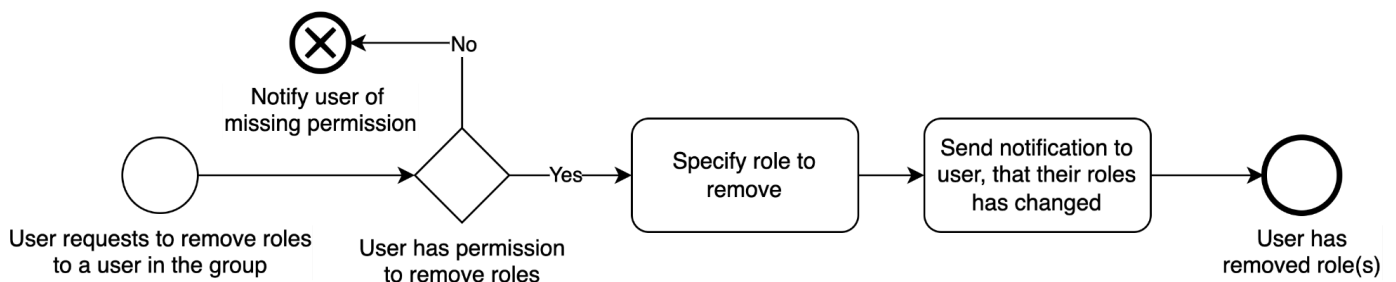
- A view of all available roles.

- The specific user's current roles.

- The user has permission to set roles.



-
- **User story:** As a user, I want to remove a user's roles within the group, so I can limit their capabilities.
 - **Description:** Allow the user to remove roles from a user.
 - **Validation criteria**
 - The specific user's current roles.
 - The user can choose which role(s) to remove.
 - The user has permission to remove roles.



Logging

- **User story:** As a group owner, I want to have all group related activity logged, so I have evidence I can inspect if I need to.
 - **Description:** Log all group related activity.
 - **Validation criteria**
 - Event data is stored whenever a member of the group performs an action.
 - Data includes the following:
 - Action
 - Status
 - Email
 - Timestamp
- **User story:** As a group owner, I want to view all logs for a specified group, so I can inspect activity in the group.
 - **Description:** Allow the group owner to view logs for the specified group.
 - **Validation criteria**
 - The group owner can specify the group for which they want to view the logs for.

- The logs are presented in a table, including the following data:
 - Action
 - Status
 - Email
 - Timestamp
-

Billing management (POSTPONED)

Payment

Payment

- **User story:** As a user, I want to create a payment option, so I can decide how I want to pay.
 - **Description:** Allow the user to create a payment option.
 - **Validation criteria**
 - Valid credentials are entered.
 - The user can have multiple payment options.
-
- **User story:** As a user, I want to view all payment options, so I can manage them properly.
 - **Description:** Allow a user to view their payment options.
 - **Validation criteria**
 - The user can view their created payment options.
-
- **User story:** As a user, I want to edit a payment option, so I can update it when needed.
 - **Description:** Allow the user to edit a payment option.
 - **Validation criteria**
 - The user can edit a payment option's information.
-
- **User story:** As a user, I want to delete a payment option, so I can remove unneeded information.
 - **Description:** Allow the user to delete a payment option.
 - **Validation criteria**
 - The payment option is deleted from all systems.
-
- **User story:** As a user, I want to specify the default payment option used for each individual group, so I can separate billing and ensure group members use the correct billing.
 - **Description:** Allow the user to specify the default payment option used for each individual group.

- **Validation criteria**

- The user has permission to set the default payment option for the group.
- When creating a case, it defaults to the specified payment option.

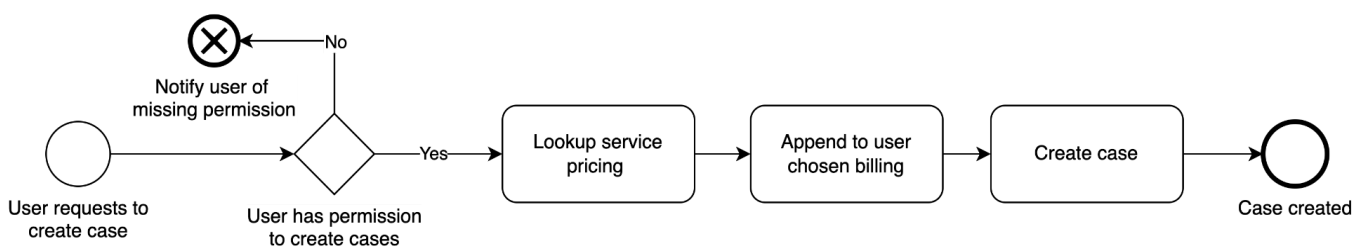
Case management

- **User story:** As a user, I want to create a case, so I can begin my work.

- **Description:** Allow a user to create a case.

- **Validation criteria**

- The user can specify the name of the case.
- The user can specify which group the case belongs to.
- The user can choose case specific parameters.
 - E.g. implementation group.
- The user can choose which billing the case should be billed.



- **User story:** As a user, I want to delete a case, so I can remove unused cases.

- **Description:** Allow a user to delete a case.

- **Validation criteria**

- The user can delete the specified case.

- **User story:** As a user, I want to edit a case's information, so I can ensure it reflects what it contains.

- **Description:** Allow the user to edit a case's metadata.

- **Validation criteria**

- The user can provide a new name for the case.

Application integration capabilities

Application embedding

Application embedding

- **User story:** As a developer, I want to embed an application, so I can create a seamless user experience when navigating from the portal.
 - **Description:** Allow the developer to embed an application.
 - **Validation criteria**
 - The application is navigable from the portal.
 - The application is fully functional when accessed from the portal.
-
- **User story:** As a developer, I want to receive relevant information from the portal when needed, so I can ensure the correct data is fetched into the application.
 - **Description:** Allow the developer to receive relevant information from the portal.
 - **Validation criteria**
 - The portal can send relevant information to the embedded application.
 - userId
 - groupId
 - case name
 - etc.
 - The embedded application can receive information from the portal.
-
- **User story:** As a developer, I want to send relevant information to the portal when needed, so I can utilize existing functionality of the portal.
 - **Description:** Allow the developer to send relevant information to the portal.
 - **Validation criteria**
 - xxx, not sure about this story..

Customer support features (POSTPONED)

- **User story:** As a user, I want to view a list of frequently asked questions, so I can seek help in case my issue is present there.
 - **Description:** Allow the user to view a list of FAQs.
 - **Validation criteria**
 - The user can view a list of frequently asked questions.
-

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, if they doesn't use any third party providers.

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.

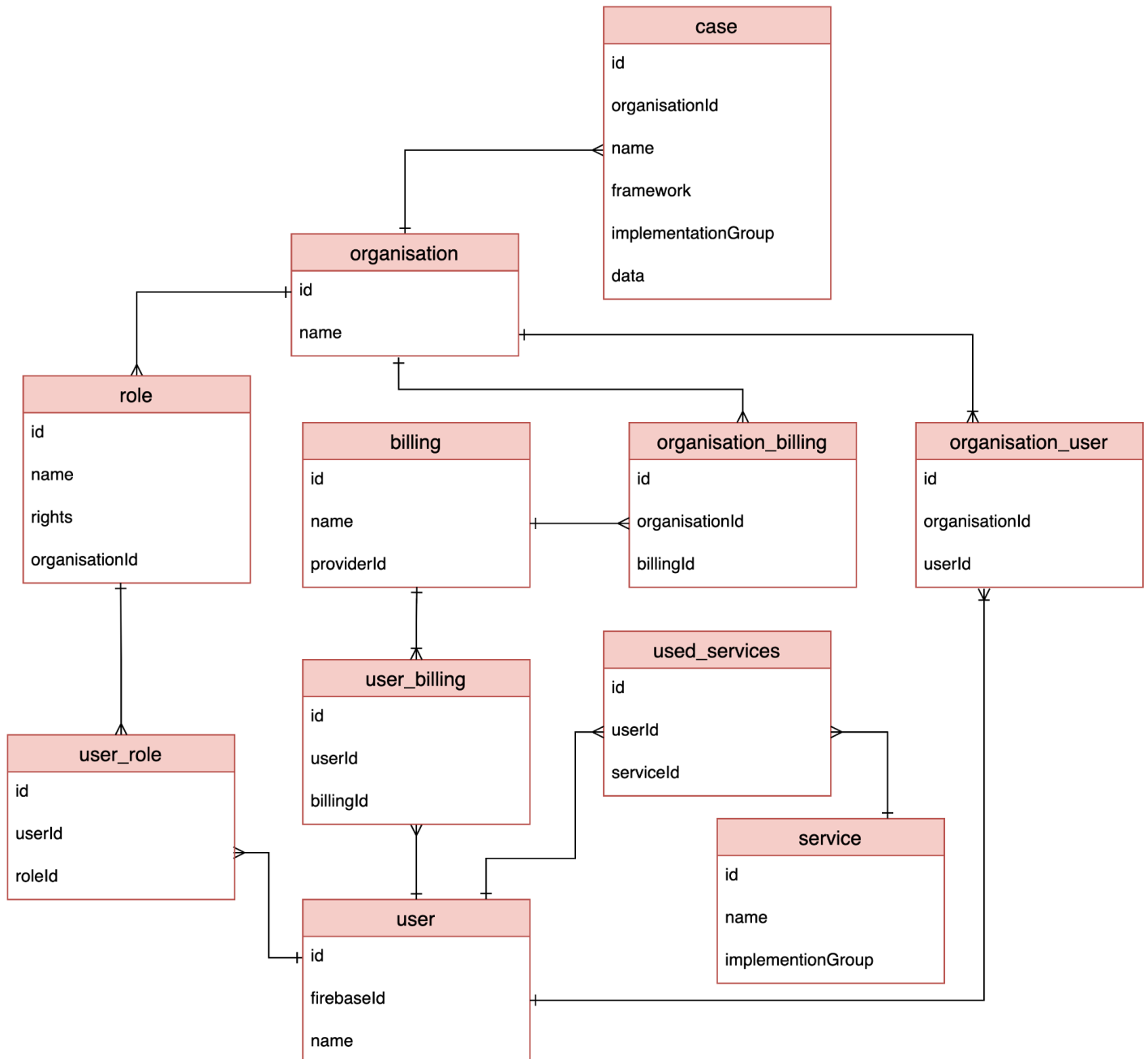
Theme

The interface must adhere to a standardized theme color palette.

Data requirements

The data structure primarily revolves around having multiple users being able to access and perform CRUD on the same resources, all while having an underlying and flexible RBAC like role management system.

Resources in the system are *users*, *roles*, *billing*, *cases*, *services*.



Appendix

Changelog

Functional requirements

- As an administrator, I want to log each individual user's actions, so I can document that our system isn't at fault if certain things happen, while also providing this service to our customers.