**Password Safe: Requirement Analysis**

**Description**

Our app saves passwords. When the user opens it for the first time, he is prompted to enter a password. Any other time the user opens the app, it will ask for that password before continuing. If the user fails to login after a certain number of attempts, the app cannot be used until a certain time. In the home page is a list of all the passwords where the name of the website and the credential level are shown. From there, the user can click any credentials, the app will then popup a message asking the user if it is safe to show the credentials, if the user agrees the credentials are shown. When it comes to saving a new credential, there will be a button which will open a new page. Credentials will be saved into three levels. The first level is just the username and password. The next level adds the email address. The last level is for if the user wants to save security questions as well. Every time the user wants to go to the next level, he has to press a button to expand the form. He can also go back. A user can also modify or delete a credential by holding a credential in the home page.

**Functional Requirements**

The application must require a secure combination to be accessed.

The application must save data by encrypting it.

The application must be able to save login information.

The application must allow credentials to be modified.

The application must allow credentials to be deleted.

**Non-Functional Requirements**

The application should lock if the combination fails.

The application should have different combination options.

The application should allow fields to be added to the credentials.

The application should allow credentials to be sorted.

The application should allow credentials to be filtered.

The application should allow the user to name the credentials.

The application should allow any number of credential-related questions to be saved.

The application should store backup

**Use-cases**

Title: Save a new credential with email address

Primary Actor: User

Success Scenario:

1. User starts the app
2. User successfully logins
3. User create a new credential
4. Enters a username
5. Enters a password
6. Selects he wants to go to the next level
7. Enters an email address
8. Saves
9. The credential is encrypted
10. The encrypted credential is saved into the database

Title: Locked from the application

Primary Actor: User

Success Scenario:

1. User starts the app
2. User enters the combination
3. Combination is invalid
4. User tries again a couple times
5. App is locked
6. User waits for a certain amount of time before trying the combination again

Title: Delete a Credential

Primary Actor: User

Success Scenario:

1. User starts the app
2. User successfully logins
3. User selects a credential he wants to delete
4. The credential is deleted from the database

Title: Modify a Credential

Primary Actor: User

Success Scenario:

1. User starts the app
2. User successfully logins
3. User selects a credential he wants to modify
4. User modifies any information he wants
5. The credential is updated in the database

**User-stories**

As a user, I want to be able to save my usernames & passwords so I can look at them later.

As a user, I want to be able to change my safe combination so that it can be more secure.

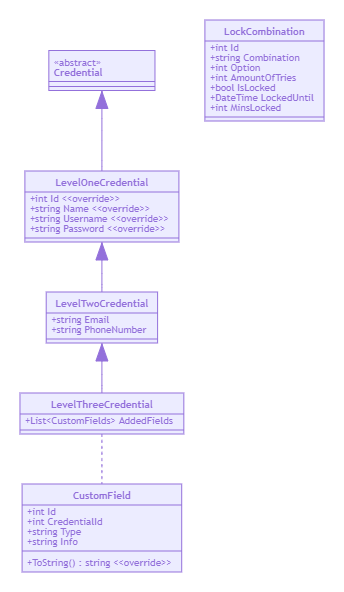
As a user, I want to be able to specify which type of credentials I am saving.

As a user, I want to be able to sort all my credentials so I can see the most recent ones.

As a user, I want to be able to search for a specific credential in order to save time.

As an owner, I want the credential information to be encrypted so that it can be more secure.

**UML class diagrams**

****

**classDiagram**

**LevelOneCredential <|-- LevelTwoCredential**

**LevelTwoCredential <|-- LevelThreeCredential**

**Credential <|-- LevelOneCredential**

**LevelThreeCredential .. CustomField**

**class Credential**

**<<abstract>> Credential**

**class LevelOneCredential{**

**+int Id <<override>>**

**+string Name <<override>>**

**+string Username <<override>>**

**+string Password <<override>>**

**}**

**class LevelTwoCredential{**

**+string Email**

**+string PhoneNumber**

**}**

**class LevelThreeCredential{**

**+List<CustomFields> AddedFields**

**}**

**class CustomField{**

**+int Id**

**+int CredentialId**

**+string Type**

**+string Info**

**+ToString() string <<override>>**

**}**

**class LockCombination{**

**+int Id**

**+string Combination**

**+int Option**

**+int AmountOfTries**

**+bool IsLocked**

**+DateTime LockedUntil**

**+int MinsLocked**

**}**

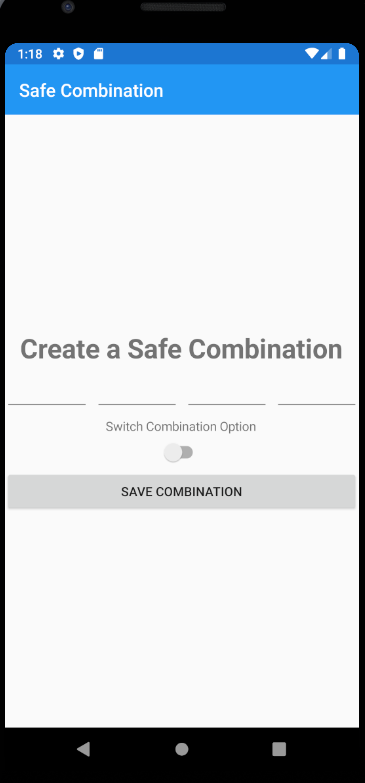
**Future Works**

* The application should store backup to the cloud.
* The application should be able to downgrade the credential level.

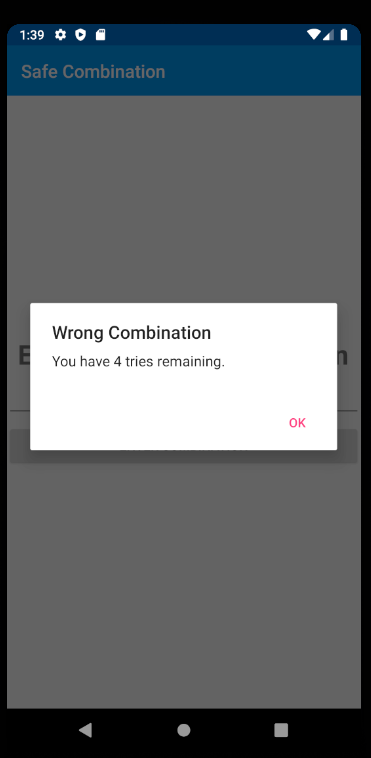
**Snapshots**

Login Page:

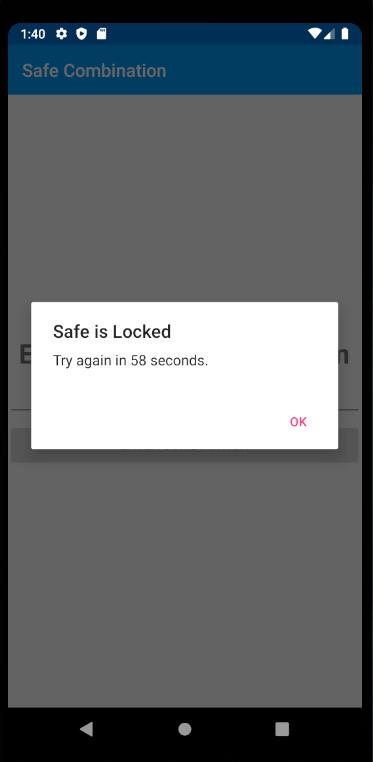
Create a credential with the switch to enter a text password instead



When entering the wrong password

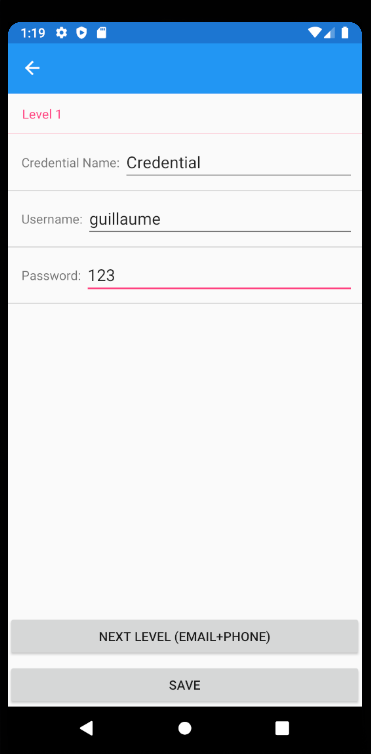


Too many wrong attempts and the app is locked

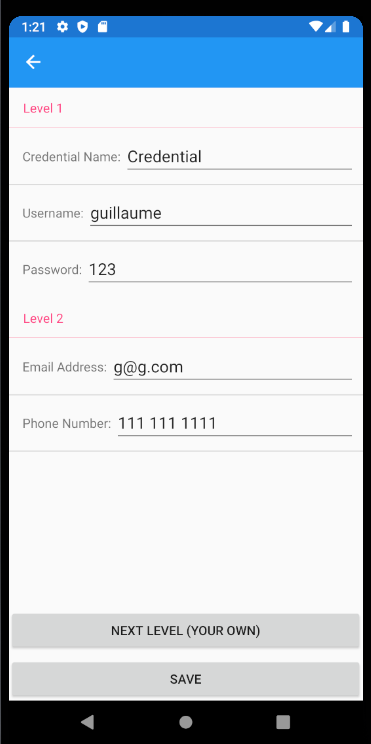


Creating/Editing a credential page:

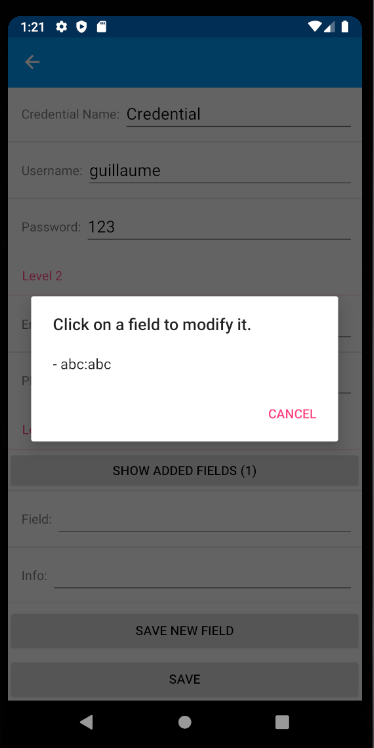
Creating a level one credential



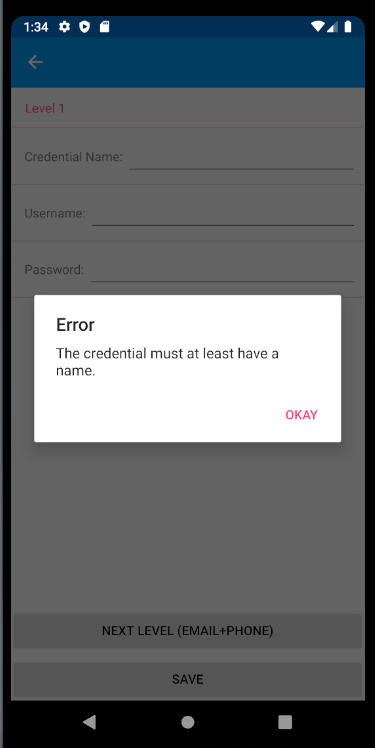
Going to a level two credential



Show the added fields, from there you can click one to either delete it or edit it

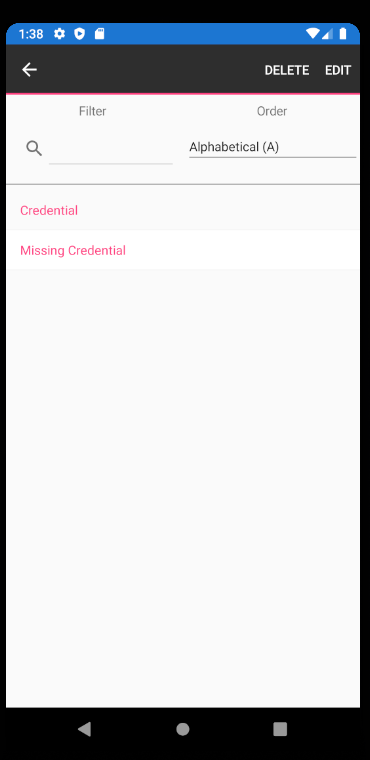


The credential requires at least a name

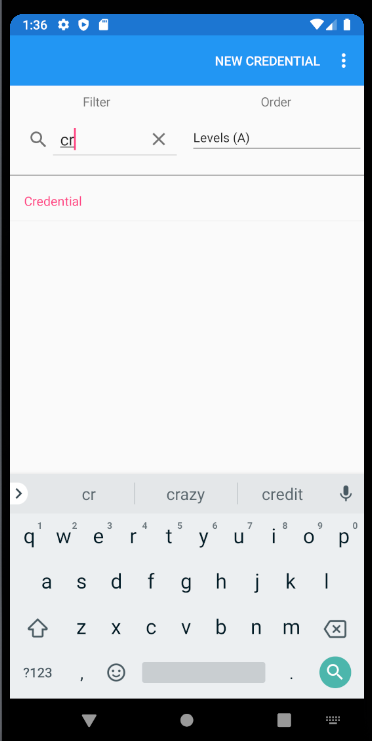


Listing credential page:

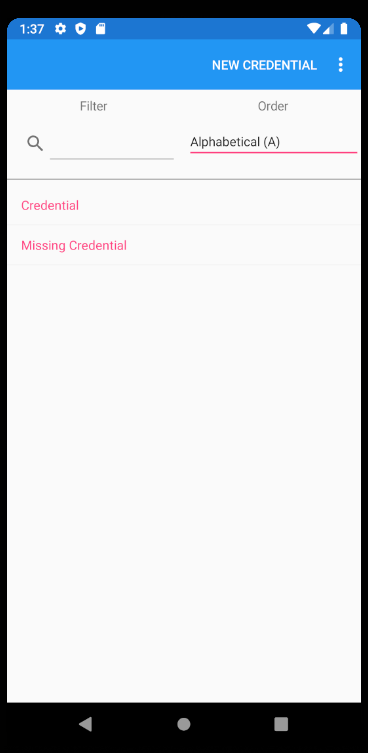
A credential can be edited or deleted if long clicked



Filtering

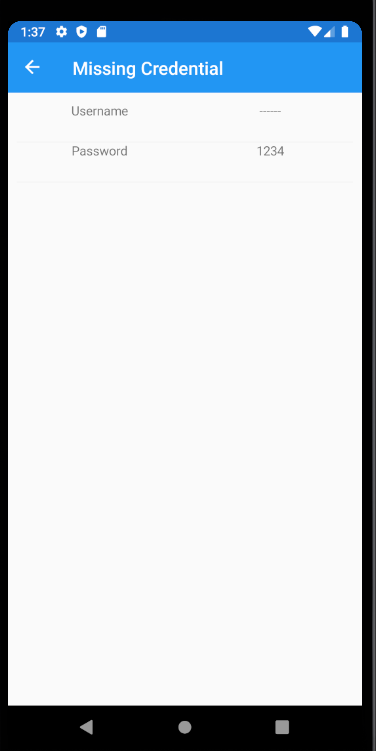


Ordering



Page showing the credential:

Demonstrating a credential where all the fields are not filled



Showing a credential filled out

