CS-360-H7351 Mobile Architect & Programming

Module 3-3: Project One

Eric Wallace

Nov. 9, 2022

The application I have chosen is the inventory application. The main goal of this application is to track items added and removed from a particular inventory. Allowing users to track the number of items that is in stock or in their possession. For a user to complete the goal of the application, all they need to do is enter an inventory.

While the application is tailored more toward personal and small business use, it could be used as a component of a much larger warehouse system. It could also be integrated with a retail system which has an online or brick-and-mortar store.

No matter what environment it is used in the application’s audience will be individuals with a need to interact with an inventory. The audience could be a user needing to track personal items located in a residence, a collector needing to keep an inventory of items in a collection, a small retail business owner needing to track store inventory or a large business wanting to track assets available for distribution to employees or track company assets that have been distributed to employees.

The screens necessary for the application would be login, a profile screen where the user could view and edit personal information, a screen for showing items, screen that will be used for adding, editing, and removing items on the system. A settings screen for used to set various global application settings. A screen that would access the camera making it possible to scan the barcode on an item either for searching for that item or adding an item to the system.

The major components and application requirements are shown below listed by screen. The database which consists of two different data structures, either noSQL or RDBMS databases, will be hosted in the cloud under a cloud platform to ensure proper security is maintained, 99.9% uptime and for deployment of application updates or upgrades. The application would communicate with the data structures via an API, this follows industry standards and guidelines. Since I have experience in Flutter, which uses some Android components, I know the components and features shown below may include items to which we have yet to cover in this course. I believe those components would be the best choice in successfully developing this application.

* **All Screens:**  Elements – TextView, two ImageButtons (search, camera and avatar) and a SearchView
  + Title bar title – TextView is used for the title which displays the title of the screen or item name on the details screen.
  + Search – SearchView is used for when users want to search for items by name. Pressing the search button will call a function that finds all items in the Items collection that contains the word or phrase entered.
  + Camera button – ImageButton is used for when a user would like to search by barcode. Pressing the button will take users to the scan screen
  + Avatar button – ImageButton is used to show the avatar of the user but also when pressed takes the user to their profile page
* **Login Screen:** Elements – Two EditText, a Button and a constraint layout

The data flow would take place in the form of the application sending the credentials to the API server via SSL for authentication which then would respond with the response. If successful the user information would be in the payload of the response from the API at which time the application would navigate to the home screen.

* **Home Screen:** Elements – Custom ListView and FloatingActionButton
  + ListView shows items in inventory, the items it displays depends on state, if it is the default view it will display all items, if a search is performed it will display the results of the search. Each item in the ListView will contain an image of the item, the title of the item with a brief description and the number of items in the inventory. Pressing and item will cause the application to switch screen in which the item number is passed to the details screen.
  + FloatingActionButton – Used for adding new items to the inventory, pressing this will switch the details page but does not pass an item number
* **Details Screen:** Elements – Multiple EditTexts and two Buttons
  + EditTexts – Used for displaying each property of an item
  + Delete Button – Used for deleting an item from inventory, pressing this button will initiate a Delete request with the API server which will remove the item from inventory
  + Save Button – Used for updating or adding items to the inventory, when pressed it sends an API call to either add or update an item in the inventory. If item number exists then a PUT request is sent to update the information and if it does not a POST request is sent to add the item.

After a save or delete action takes place the home screen will be shown along with a notification the item was either updated, created or deleted.

* **Settings Screen:** Elements – Multiple EditTexts and a Button
  + EditTexts – Used for displaying each property of an item
  + Save Button – Used for updating settings for the application, pressing the save button initiates the application to save the settings for the application locally on the device

After a save operation is complete the user is navigated back to the home screen and a notification is shown settings were updated

* **Profile Screen:** Elements – Multiple EditTexts, an ImageButton and a Button
  + EditTexts – Used for displaying each property of an item
  + ImageButton – Used for displaying the avatar of the user, when pressed it will open a file dialog allowing the user to select an image that is located on the device.
  + Save Button – Used for updating the users profile, when pressed it sends a PUT request to the server to save the updated information.

After a save takes place the home screen will be shown along with a notification the profile was either updated

* **Scan Screen:** Elements: ImageView

ImageView – Used for showing the image being shown by the camera, whenever the camera successfully scans a barcode the data is then transmitted back to the screen that requested it.