Assignment: Pocket Store

Objectives:

Create a mobile app using Ionic and Firebase for Store Item Management.

Instructions:

1. This is an **individual** assignment.
2. The assignment is **100 marks** and constitutes 4**0%** of the total module marks.
3. **Late** submission will be **penalized**.

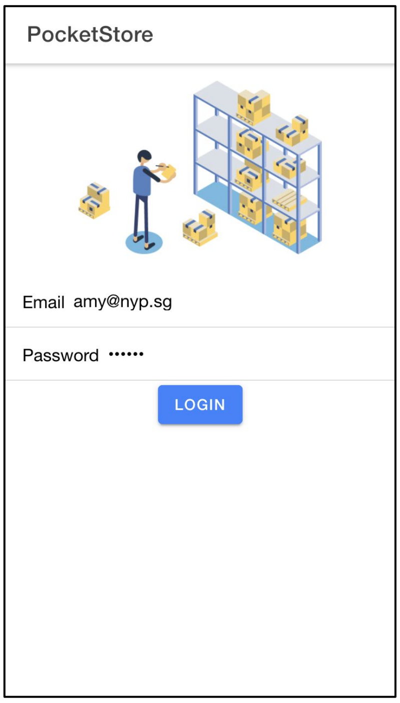
a) Cap at 50% of base mark for submission within 5 calendar days from deadline
b) From 6th day within deadline onwards, award 0 mark
c) Example:
  i) Deadline is 1 Dec, Sunday 2359, base mark is 100 marks
  ii) Submission is from 2 Dec to 6 Dec (inclusive)
  iii) If learner scored 80 marks, a score of 50 marks will be awarded.
  iv) If learner scored 40 marks, a score of 40 marks will be awarded.
  v) Submission is from 7 Dec onwards, a score of 0 marks will be awarded.

Submission:

Brightspace Submission	Zip file of source code (`src` folder) Supply readme file of installation steps if you have used other frameworks other than firebase. Any updates to codes after the demo strictly NOT allowed unless authorised by tutor. If found, NO additional marks for the new updates in the submitted source.	Week 15 (**3 Aug 2025 2359**)
Demo & Presentation	Run your Ionic App and demo all the app features. You can show in browser if not using device functionalities. Show on simulator or real phone if need to demo device functionalities. Shows that your data in the Firebase is updated as your app performs its functionality Max 5 mins(failure to present will get penalty of 50%) Technical walkthrough on source codes in **week 14** is compulsory.	Week 15

1 Beginner

There are 6 pages in the basic requirements of this mobile app.

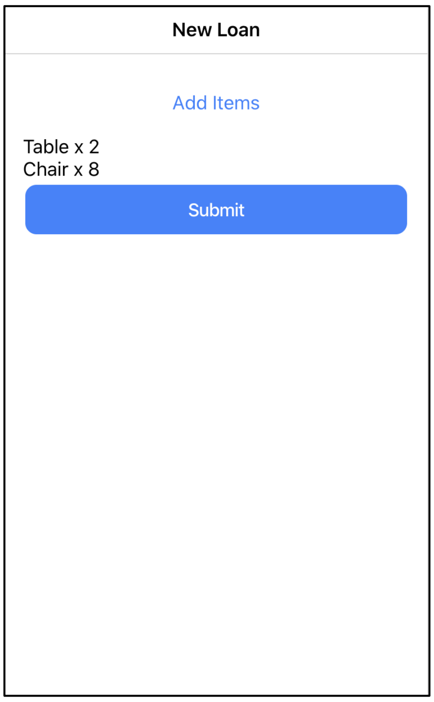

PocketStore

Email amy@nyp.sg

Password ••••••

LOGIN

Login Page

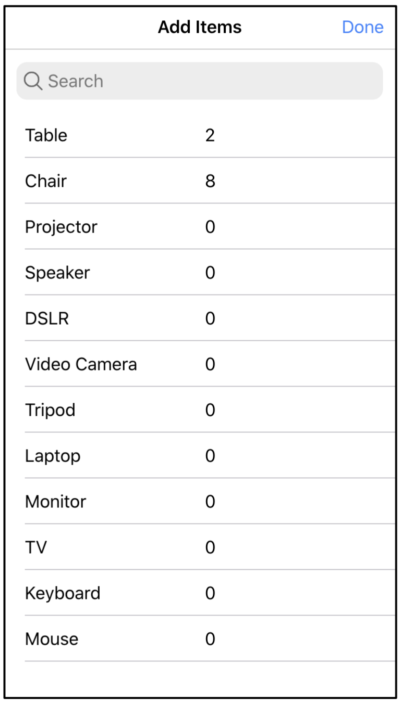

New Loan

Add Items

Table x 2
Chair x 8

Submit

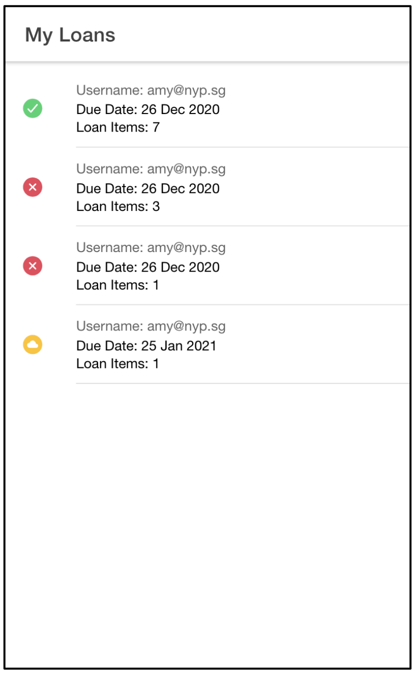
New Loan Page


Add Items Done

Search

Table	2
Chair	8
Projector	0
Speaker	0
DSLR	0
Video Camera	0
Tripod	0
Laptop	0
Monitor	0
TV	0
Keyboard	0
Mouse	0

Add Items Page


My Loans

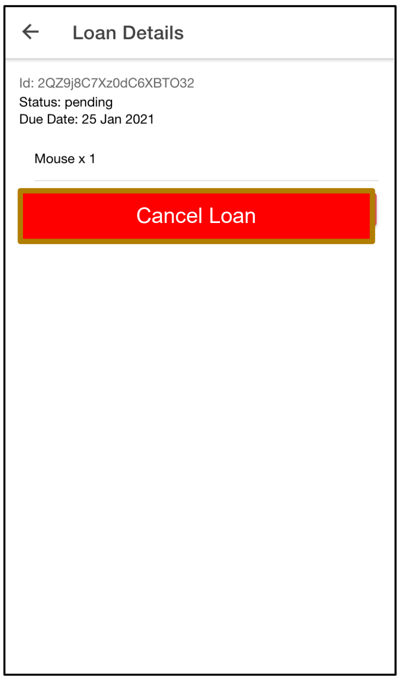
✓ Username: amy@nyp.sg
Due Date: 26 Dec 2020
Loan Items: 7

✕ Username: amy@nyp.sg
Due Date: 26 Dec 2020
Loan Items: 3

✕ Username: amy@nyp.sg
Due Date: 26 Dec 2020
Loan Items: 1

● Username: amy@nyp.sg
Due Date: 25 Jan 2021
Loan Items: 1

Loans Page

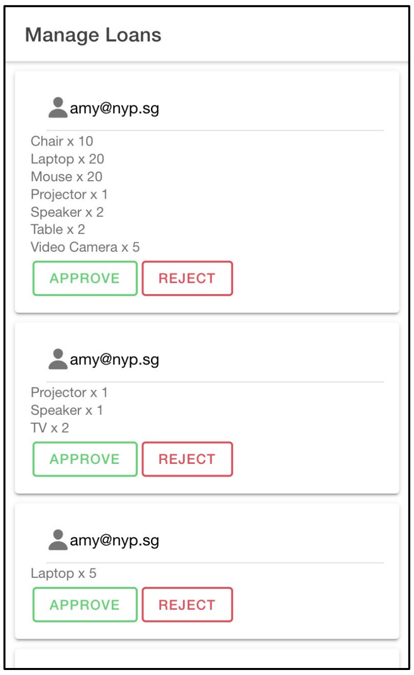

← Loan Details

Id: 2QZ9j8C7Xz0dC6XBTO32
Status: pending
Due Date: 25 Jan 2021

Mouse x 1

Cancel Loan

Detail Page


Manage Loans

amy@nyp.sg

Chair x 10
Laptop x 20
Mouse x 20
Projector x 1
Speaker x 2
Table x 2
Video Camera x 5

APPROVE REJECT

amy@nyp.sg

Projector x 1
Speaker x 1
TV x 2

APPROVE REJECT

amy@nyp.sg

Laptop x 5

APPROVE REJECT

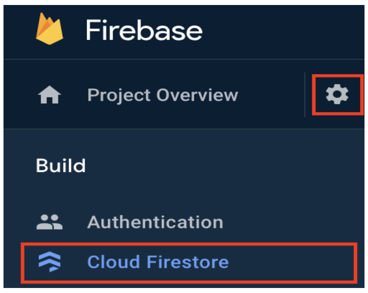
Manage Page

1.1 Database

You must connect the Ionic App to your own Firebase project.

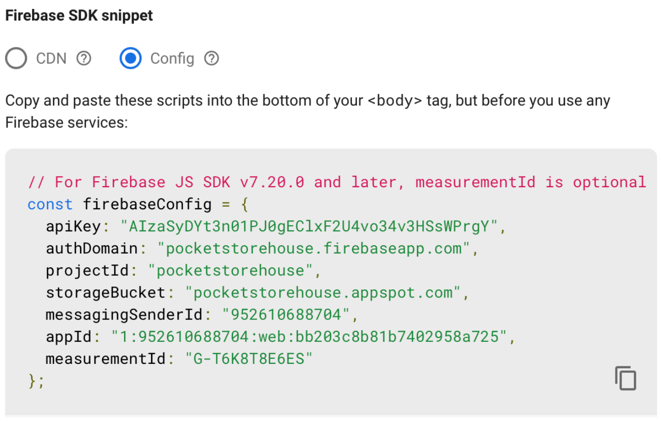
Create a new **Firebase** project.

Enable the Database **Cloud Firestore** to start in *test mode*.



Go to **Project Settings** and add a new **Web** app.

Copy the config files to your Ionic project. Please replace the firebaseConfig from the startup file with yours.


Firebase SDK snippet

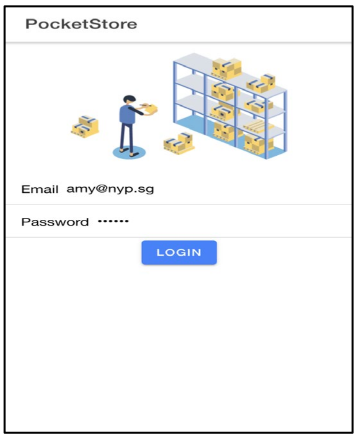
CDN [✓ Config]

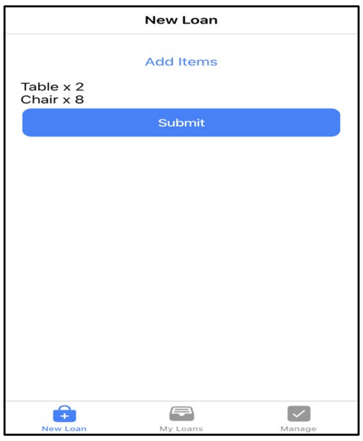
Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

```
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyDYt3n01PJ0gEClxF2U4vo34v3HSsWPrgY",
  authDomain: "pocketstorehouse.firebaseapp.com",
  projectId: "pocketstorehouse",
  storageBucket: "pocketstorehouse.appspot.com",
  messagingSenderId: "952610688704",
  appId: "1:952610688704:web:bb203c8b81b7402958a725",
  measurementId: "G-T6K8T8E6ES"
};
```

1.2 Login

When the app launches, the **Login Page** is shown.There is no need to enter email and password. Simply click on **Login** button to go to the **New Loan Page**. **New Loan Page** shows on the first tab of three – *New Loan*, *My Loans* and *Manage*.





1.3 Create a new Loan

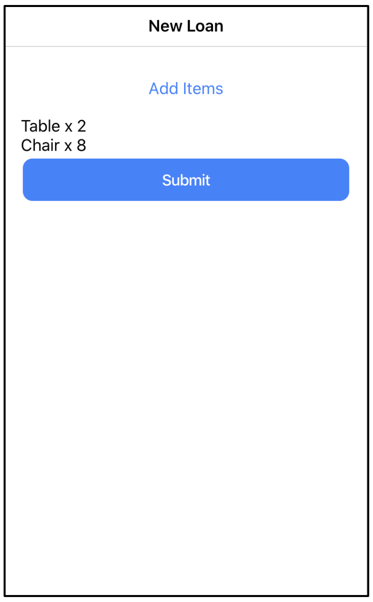
From **New Loan Page**, click Add Items to go to **Add Items Page**.

Search for the item and edit the quantity. Click **Done** to go back to **New Loan Page**.

Verify your loan items or go to **Add Items Page** to modify your items. Use Services to share data between the 2 pages.

When done, click **Submit** to create a new loan in the Database. A new loan is created with `status` set as *'pending'* and the `duedate` set 2 weeks from today's date. A **Toast** message shows indicating the new loan `id` when the Database operation completes successfully.

After submitting, the quantity of ALL the loan items on both **Add Items Page** and **New Loan Page** should be reset to quantity 0 for user to create the next loan. **Ensure Search bar is able to search from the list.**



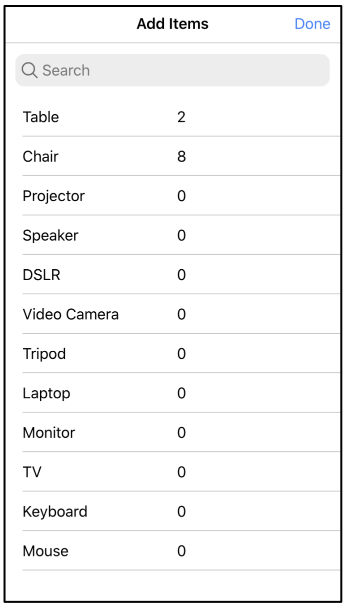


Table	2
Chair	8
Projector	0
Speaker	0
DSLR	0
Video Camera	0
Tripod	0
Laptop	0
Monitor	0
TV	0
Keyboard	0
Mouse	0

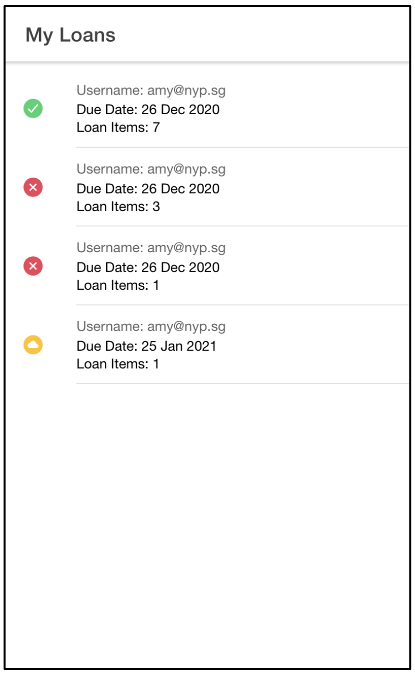
1.4 Retrieve Loans

In **Loans Page**, display the loans from the Database sorted by `duedate`, with the nearest due date displayed at the top. The due date is displayed in the format *'dd MMM yyyy'*.

To convert Firestore Timestamp to JS **Date()** object use Firestore `.toDate()` and use Date pipe to format.

- A green icon *'checkmark-circle'* shows when the loan status is *'approved'*.
- A red icon *'close-circle'* shows when the loan status is *'rejected*.
- A yellow icon *'cloud-circle'* shows when the loan status is *'pending*.

Clicking on the loan item will go to **Detail Page** showing all the loan data and items.

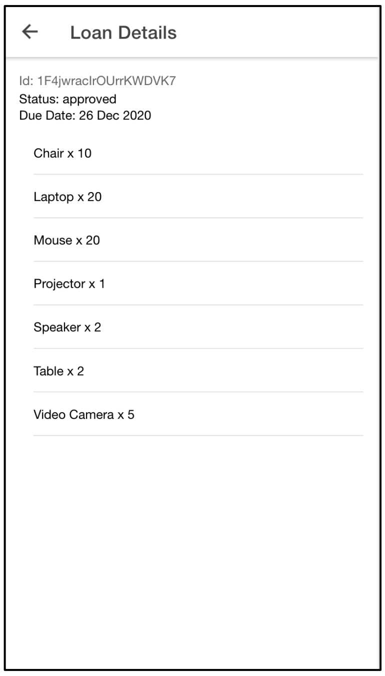

My Loans

Username: amy@nyp.sg
Due Date: 26 Dec 2020
Loan Items: 7

Username: amy@nyp.sg
Due Date: 26 Dec 2020
Loan Items: 3

Username: amy@nyp.sg
Due Date: 26 Dec 2020
Loan Items: 1

Username: amy@nyp.sg
Due Date: 25 Jan 2021
Loan Items: 1


Loan Details

Id: 1F4jwraclrOUrrKWDVK7
Status: approved
Due Date: 26 Dec 2020

Chair x 10

Laptop x 20

Mouse x 20

Projector x 1

Speaker x 2

Table x 2

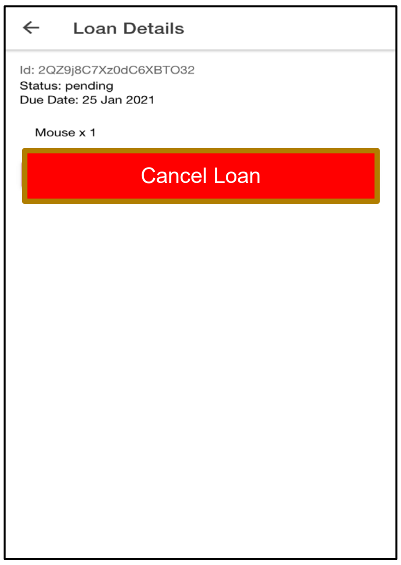
Video Camera x 5

1.5 Delete Pending Loan

If the loan has *'pending'* `status`, the **Detail Page** will show a **Cancel Loan** button.

Click on the **Cancel Loan** button to delete the loan from the Database.

Verify that the cancelled loan does NOT show on **Loans Page** anymore.

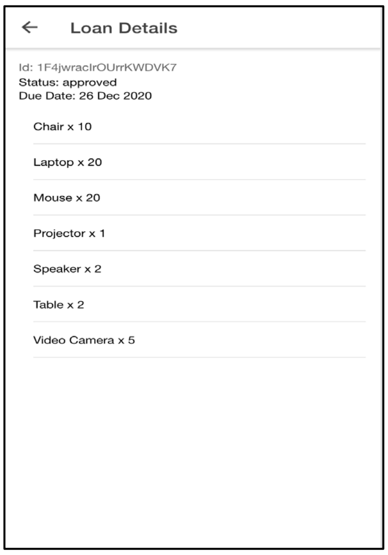


Loan Details

Id: 2QZ9j8C7Xz0dC6XBTO32
Status: pending
Due Date: 25 Jan 2021

Mouse x 1

Detail Page of *'pending'* loan shows **Delete Loan** button



Loan Details

Id: 1F4jwraclrOUrrKWDVK7
Status: approved
Due Date: 26 Dec 2020

Chair x 10
Laptop x 20
Mouse x 20
Projector x 1
Speaker x 2
Table x 2
Video Camera x 5

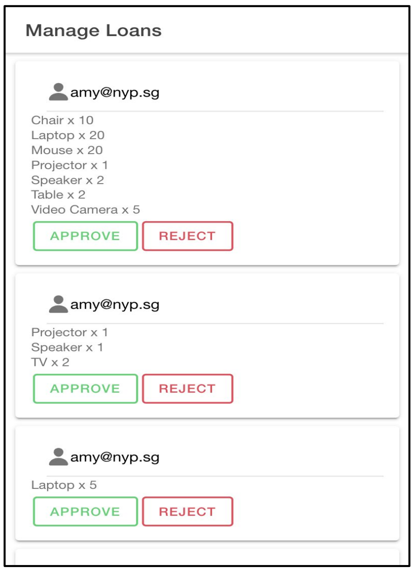
Detail Page of *'approved'* loan does NOT show **Cancel Loan** button

1.6 Manage Loans

In **Manage Page**, display the loans from the Database.

- Click **Approve** button to change the loan `status` to *'approved'* in the Database.
- Click **Reject** button to change the loan `status` to *'rejected'* in the Database.

Verify that the loan status is changed in **Loans Page**.



Manage Loans

amy@nyp.sg
Chair x 10
Laptop x 20
Mouse x 20
Projector x 1
Speaker x 2
Table x 2
Video Camera x 5

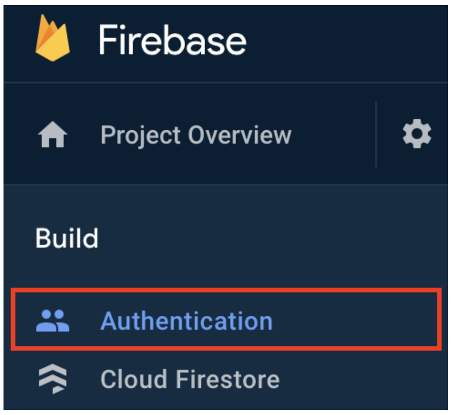
amy@nyp.sg
Projector x 1
Speaker x 1
TV x 2

amy@nyp.sg
Laptop x 5

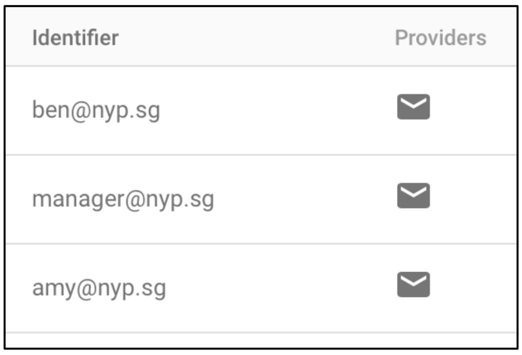
2 Intermediate

2.1 Authentication

Enable **Firebase Authentication** by **Email**.



Add 3 users with the following emails and the password `123456`.

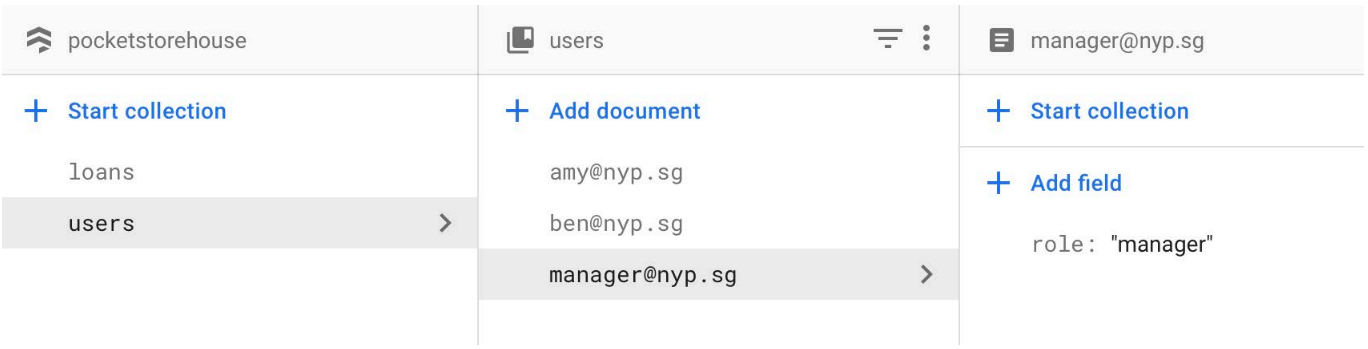


Identifier	Providers
ben@nyp.sg	✉
manager@nyp.sg	✉
amy@nyp.sg	✉

You can add other users, but your tutor will only test your app with these 3 users.

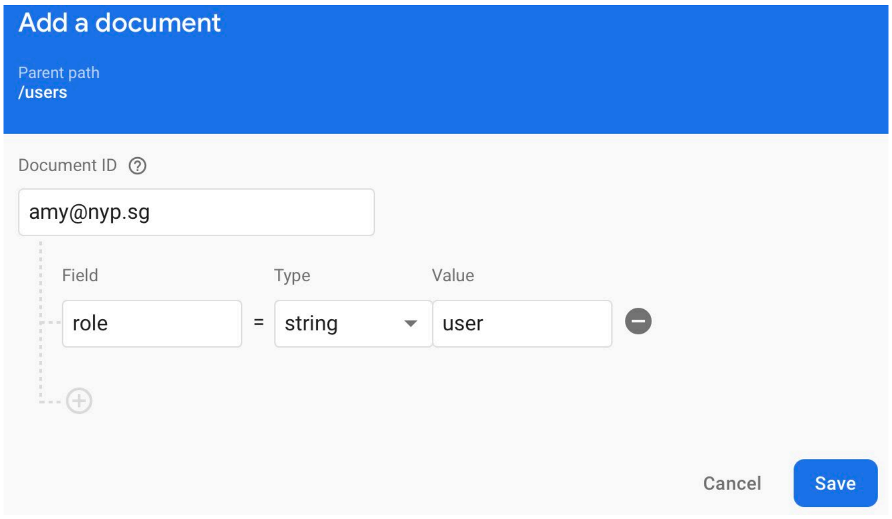
2.2 User Role

Add *'users'* collection to the **Firestore** with 3 documents.



role: "manager"

Only manage@nyp.sg has *'manager'* role. The other users have *'user'* role.


Add a document

Parent path
/users

Document ID
amy@nyp.sg

Field	Type	Value
role	string	user

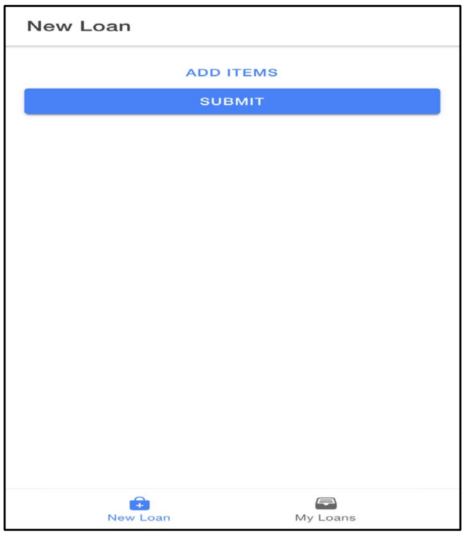
Cancel Save

2.3 Different Landing Page

For simplicity, the mobile app only has login feature. To logout, simply close the app. All the user accounts will be **created by administrators** in the Firebase console and **users cannot sign up** with the mobile app.

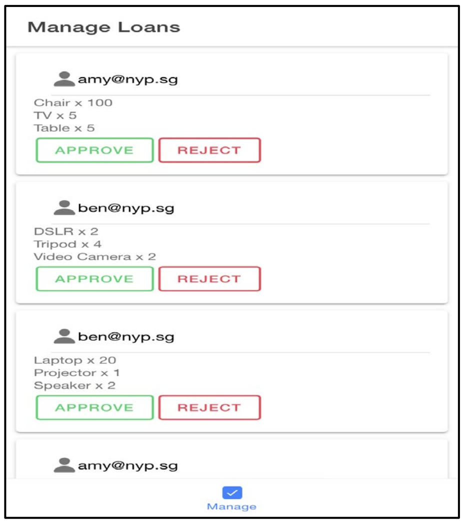
After login, there is a different landing page based on the **user role stored** in Database.

When logged in with the *'user'* role, the mobile app shows two tabs – *New Loan* and *My Loans*.



Landing Page for *'user'* role – **New Loan Page**

When logged in with the *'manager'* role, the mobile app shows one tab – *Manage*.


Manage Loans

amy@nyp.sg
Chair x 100
TV x 5
Table x 5
APPROVE REJECT

ben@nyp.sg
DSLR x 2
Tripod x 4
Video Camera x 2
APPROVE REJECT

ben@nyp.sg
Laptop x 20
Projector x 1
Speaker x 2
APPROVE REJECT

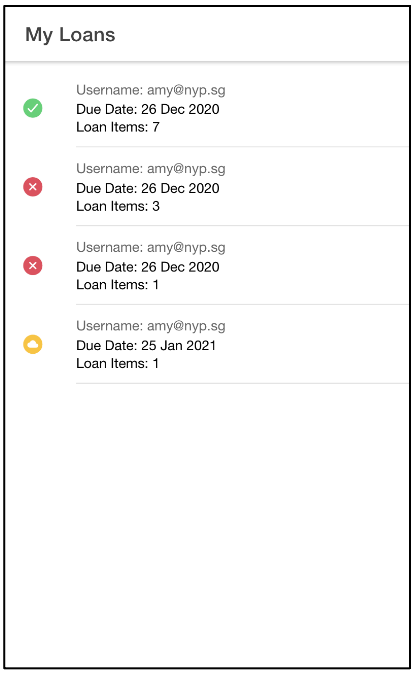
amy@nyp.sg

Manage

Landing Page for *'manager'* role – **Manage Page**

2.4 Show My Loans

Get the `username` of the logged in user. In **Loans Page**, only retrieve the loans belonging to that user.



My Loans

✓ Username: amy@nyp.sg
Due Date: 26 Dec 2020
Loan Items: 7

✕ Username: amy@nyp.sg
Due Date: 26 Dec 2020
Loan Items: 3

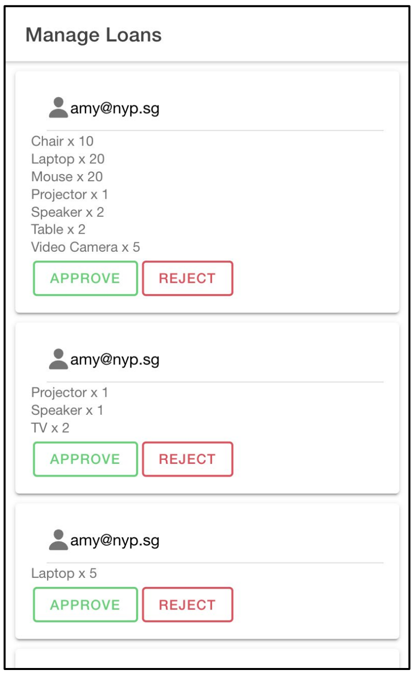
✕ Username: amy@nyp.sg
Due Date: 26 Dec 2020
Loan Items: 1

Username: amy@nyp.sg
Due Date: 25 Jan 2021
Loan Items: 1

2.5 Show Pending Loans

In **Manage Page**, show only loans with ‘*pending*’ `status`.

Hint: Use `.where()` in your Firestore query.



Manage Loans

amy@nyp.sg
Chair x 10
Laptop x 20
Mouse x 20
Projector x 1
Speaker x 2
Table x 2
Video Camera x 5
APPROVE REJECT

amy@nyp.sg
Projector x 1
Speaker x 1
TV x 2
APPROVE REJECT

amy@nyp.sg
Laptop x 5
APPROVE REJECT

3 Advanced

Add enhancements to the Pocket Store mobile app. Be innovative and enterprising and propose your own improvements which is a continuation of the Intermediate source. You may propose your innovative idea by submitting the idea in **Brightspace forum before term break** **by 10 Jun 2025**. If an innovation is identical or similar, marks will be given to the first proposer. Each learner can propose max of 2 ideas. If more than 2 ideas proposed, only first 2 ideas will be considered.

Examples

- Add stock level to each Loan Item. User cannot loan items with quantity more than the existing stock level. When user loan item successfully, the available stock level decreases. If user cancels the loan, the available stock level increases. Developer needs to create stock level for each item at the start of the mobile app launch with data in the firebase at code level NOT manually configured at firebase.

4 Grading Rubrics

Criteria	Excellent (A)	Good(B)	Satisfactory(C)	Needs Improvement (D)	Unsatisfactory(F)
Technical walkthrough **(30 marks)**	24.0-30 Completed advanced codes, running without errors, with comprehensive comments. Fully explains ALL parts of the app, clearly stating Gen AI codes(if used) during the walkthrough.	21.0-23.9 Started advanced codes but not fully completed. Able to run parts of advanced features. Able to explain codes written without the aid of Gen AI	18.0-20.9 Completed both Beginner and Intermediate levels. Able to explain codes, with both fully running without errors.	15.0-17.9 Completed Beginner's level. Able to explain codes written without the aid of Gen AI, but only started Intermediate codes.	0-14.9 Did not complete Beginner's codes.
Beginner **(10 marks)** **Api calls to firebase must use services as taught in practical. Else will be ZERO marks**	8.0-10 Implemented a complete solution with 100% compliance to requirements and no reliance on Gen AI.	7.0 – 7.9 Implemented a solution with most code written independently, demonstrating understanding.	6.0 – 6.9 Implemented a solution where most code is generated by Gen AI, showing minimal understanding.	5.0- 5.9 Implemented a solution that runs but unable to explain the code.	0-4.9 Failed to implement a working solution. Did not comply with requirements or use services for API calls to Firebase.

Criteria	Excellent (A)	Good(B)	Satisfactory(C)	Needs Improvement (D)	Unsatisfactory(F)
Intermediate **(20 marks)** **Api calls to firebase must use services as taught in practical. Else will be ZERO marks**	16.0-20 Successfully integrated with Firebase Authentication and Firestore, performing all role-based and user-based CRUD operations flawlessly. No reliance on Gen AI, code written independently.	14.0-15.9 Successfully integrated with Firebase Authentication and Firestore, performing most role-based and user-based CRUD operations with minor issues.	12.0-13.9 Successfully integrated with Firebase Authentication and Firestore, performing some role-based and user-based CRUD operations, but with significant issues.	10.0-11.9 Minimal integration with Firebase Authentication and Firestore, performing very few role-based and user-based CRUD operations.	0-9.9 No integration with Firebase Authentication and Firestore; no role-based or user-based CRUD operations implemented.
Advanced 1 **(10 marks)** **Unique idea must be approved by staff before term break else will not qualify for advance section marking** **Front-end (UI & Interactivity)**	8.0-10 Full integration of advanced UI design principles, highly polished user interface, seamless responsiveness across all devices. Includes rich interactivity (animations, dynamic	7.0 – 7.9 Good UI design, responsive across most devices, some interactivity (e.g., animations, dynamic content), but could benefit from minor usability improvements.	6.0 – 6.9 Functional UI with basic responsiveness. Lacks advanced interactivity and dynamic elements. Could use design consistency and usability improvements.	5.0 – 5.9 UI design lacks polish and responsiveness issues, minimal interactivity, and no dynamic content.	0-4.9 UI is poorly designed, non-responsive, and lacks usability. No interactivity or dynamic elements.

Criteria	Excellent (A)	Good(B)	Satisfactory(C)	Needs Improvement (D)	Unsatisfactory(F)
	content, real-time feedback) for exceptional UX.				
Advanced 2 **(20 marks)** **Back-end (Firebase Integration, Security, and Performance)**	16.0-20 Complete Firebase integration (authentication, Firestore, Cloud Functions, etc.), with high security (well-defined Firestore rules, encryption) and optimized performance (real-time syncing, low latency, fast load times).	14.0-15.9 Strong Firebase integration with core services (auth, Firestore). All features with firebase are working well.	12.0-13.9 Basic Firebase integration (CRUD, user authentication). Some performance or security issues (e.g., basic Firestore rules, minor syncing delays).	10.0-11.9 Incomplete Firebase integration, missing core features (auth, Firestore). Performance is slow, and security measures are weak.	0-9.9 No Firebase integration, or severely incomplete setup. Critical performance, security, or functionality flaws.

Innovation solution **(10 marks)** **Unique idea must be approved by tutor by 10 Dec 2024 else will not qualify for this section marking**	8.0-10 Produces original ideas with no guidance, providing strong evidence for real-world usage.	7.0 – 7.9 Produces original ideas with some guidance, offering merit and evidence for real-world usage.	6.0 – 6.9 Produces original ideas but requires guidance; solutions have some merit and evidence for potential usage.	5.0 – 5.9 Does not produce original ideas, instead enhancing existing solutions; requires significant guidance.	0-4.9 Does not produce original ideas; solutions are unrealistic, ignoring the needs and interests of the target audience.

5 Plagiarism

- Definition of “Plagiarism” under NYP Academic Integrity Policy

  “Taking and using the whole or any part of ideas, words or works of others, including contents generated by AI tools and passing it off as one’s own work without acknowledgement of the original source.“
- Learners
  - **must not** use AI tools for plagiarism or cheating by presenting generated output without proper acknowledgement of the original source.
  - Only use the AI tools to supplement their studies, clarify ambiguous concepts, and assist with research, in an ethical and responsible manner.
  - Properly cite the source and credit the AI tools in source codes under comments and indicate to the tutor during technicalcode review
  - Be able to prove that they comprehend the output of the AI tool and know how to apply it in their academic work when being asked

~ End ~