Product Design Specification

AR Notebook

Prepared by:
Megan Majewski
Arman (Mohammed) Ali
Artur Bushi
Darion Thompson

Revision History

Date	Revision Number	Description
10/9/17	Version 1.0	Created Template for the document. Inserted content for sections 1, 1.2, 2
10/18/17	Version 1.1	Assumptions , Architecture Diagram and Use Cases
10/24/17	Version 1.2	Sequence Diagrams
10/30/17	Version 2.0	Finalized Copy
11/1/17	Version 2.1	Post-Presentation Additions

Table of Contents

Contents

Table of Figures

1 Introduction

1.1 Purpose of the Product Design Specification Document

The Product Design Specification document documents and tracks the necessary information required to effectively define architecture and system design in order to give the development team guidance on architecture of the system to be developed. The Product Design Specification Document's intended audience is the project manager, project team, and development team. Some portions of this document such as the user interface (UI) may on occasion be shared with the client/user, and other stakeholder whose input/approval into the UI is needed.

2 General Overview and Design Guidelines/Approach

This section describes the principles and strategies to be used as guidelines when designing and implementing the system.

2.1 Assumptions / Constraints / Standards

This section outlines any assumptions, constraints, or standards that are built into the development of this application. The following list of assumptions and constraints must be met in order for the application to run its full functionality.

2.1.1 User Assumptions

1. The Users will have a Facebook account.
2. The users have a stable internet connection in order to authenticate into the application.
3. The user's device will have a fully functional and responsive touch screen to interact with the application.
4. The users have a familiarity with how to use iOS Applications and are able to download applications from the app store and run these applications on their device.

2.1.2 Technical Constraints

1. This application is built exclusively for iOS devices running iOS 11 or higher with a A9 processor and a working camera.
2. The user has granted permissions for the application to access the device's camera
3. The application uses Apple's Swift, ARKit, and SceneKit internally for the application.
4. Externally, the application uses Firebase for data storage and user authentication.

2.1.3 Design Constraints

1. This application will function in a single view, apart for Facebook authentication. The user should have the ability to perform all interactions without leaving the augmented reality view.

2.1.4 Hardware Constraints

In order to use Apple's newly released ARKit the application user must have a device running iOS 11. This device must also include an A9 processor. The list of devices that support the hardware requirements are as follows:

1. iPhone 6s and 6s Plus
2. iPhone 7 and 7 Plus
3. iPhone SE
4. iPad Pro (9.7, 10.5 or 12.9) – both first-gen and 2nd-gen
5. iPad (2017)
6. iPhone 8 and 8 Plus
7. iPhone X/iPhone 10

3 Architecture Design

This section outlines the system and hardware architecture design of the system that is being built.

3.1 Hardware Architecture

Figure 1 Hardware Architecture

The application consists of four parts. The user, the mobile device, which also relies on an iSight camera, and the Firebase server. The user place notebook objects in augmented reality using the mobile device and its camera. The textual information they place on notebook pages will be stored to the Firebase server under that user's Facebook ID. When the user requests to view previous notes they have placed a request is sent from the device to the Firebase server, once the information requested is received the content is rendered on the iOS device to the user. The Firebase Realtime Database supports 100 simultaneous connections, with 1GB of stored data. We also utilize Firebase Storage to contain the uploaded gallery images which allows for 5GB of data and 1GB of downloads per day.

3.2 Software Architecture

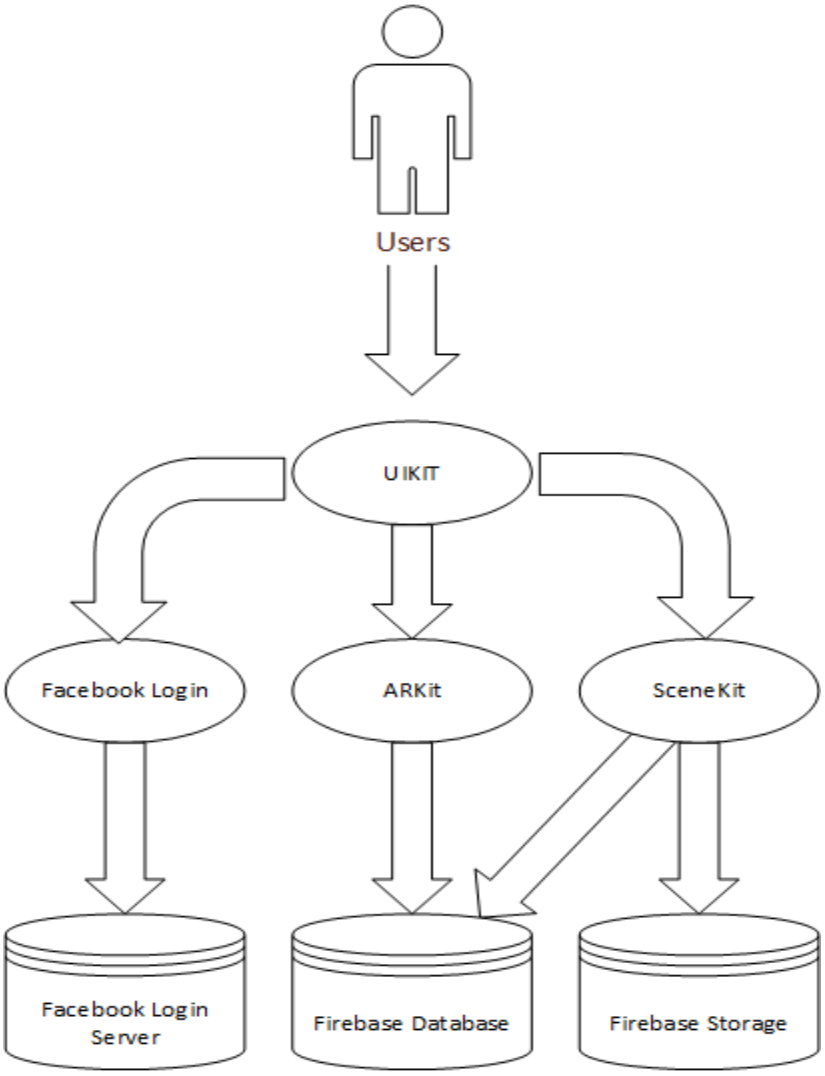


Figure 2 Software Architecture

In this diagram, there are four levels shown. The user's notebook information is saved across one server and two databases. Facebook Login server is its own entity which monitors the usage of the user's Facebook account as a means of authentication. Firebase Database is a NoSQL database which only stores items in a JSON format. Firebase Storage is a sandbox-type storage system, known as a "bucket", which stores any image. The information in the Facebook server is passed back through the LoginButtonDelegate that is implemented in the FacebookLogin Library. The Firebase Database sends the string information to SceneKit and ARKit, which work in tandem to render the model and present in UIKit. All three of these processes get molded into one view, via UIKit, which is shown on the user's iOS device screen.

3.3 Security Architecture

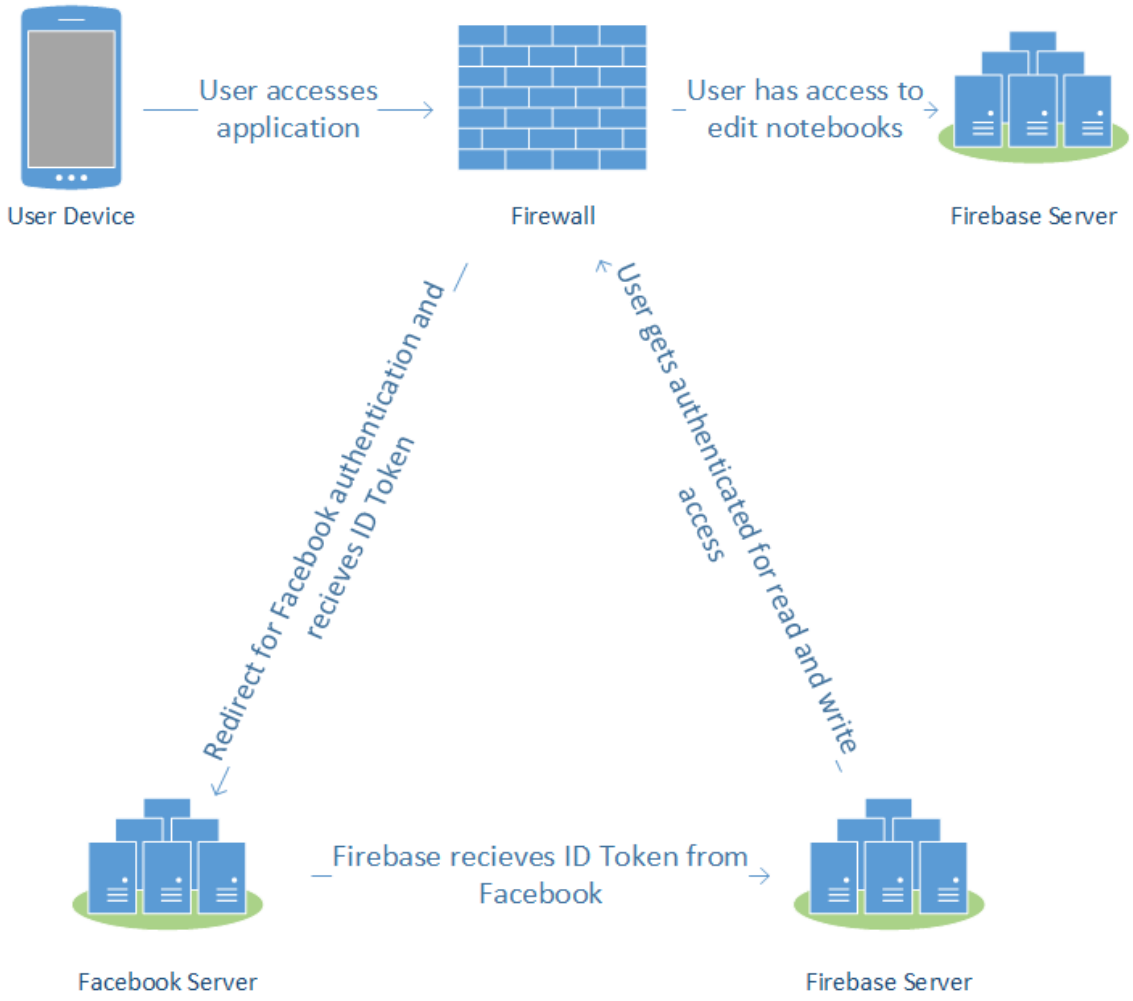
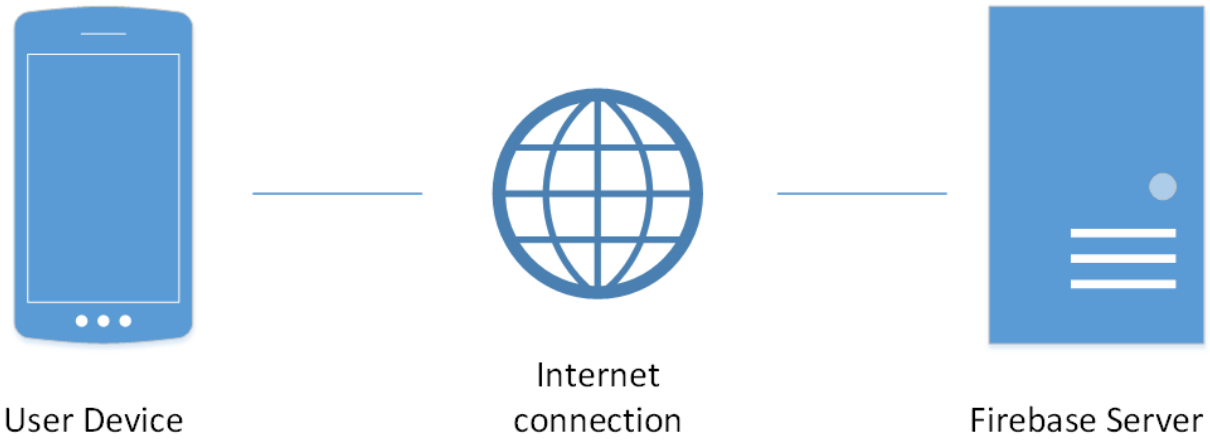


Figure 3 Security Architecture

To use any aspect of the application, the user needs to be authenticated through Facebook. The user opens the application, prompting them to either log into a valid Facebook account or to register for one. After logging in, the application sends the access token acquired from Facebook login to Firebase. Firebase then authenticates the user with that access token and grants the user read/write access to their personal notebook information on the Firebase database. Going forward, the user can change any of the available notebooks and their content.

In the AR Notebook application, all users have read and write access only when it comes to their own notebooks. There is no admin login who can go in and change the user's' information through the application itself. Services of that nature are delegated to manual Firebase access, which no user can gain access to.

3.4 Communication Architecture



The communication architecture is based off of a continuous and reliable internet connection. The users need to access both the Facebook Login server and the Firebase server to log into the application. After logging, in the user can request previous notebook content from the server.

4 System Design

4.1 Use-Cases

For our application, the primary actor for all of our use cases will be the Application User. This is because our application is intended for personal use only.

Use Case ID	UC-1 Facebook Login
Created:	10/18/2017
Actors:	Application User
Description:	This use case will describe the process a user takes to login to the application for the first time.
Trigger:	The first time the user opens the application on their device.
Precondition:	The application is installed to the Application User's device. The user has a valid Facebook account.
Postcondition:	The user is authenticated into the application the welcome screen will show. The user can follow the rest of the use cases below.
Normal Flow	1. Open the application. 2. Tap the Login with Facebook Button. 3. Enter valid credentials. 4. Grant permission the AR Notebook application to access Facebook profile.
Alternative Flow	1. Open the application. 2. The user is previously logged in with Facebook on this device. 3. The application goes directly to the welcome screen skipping the login.
Frequency of Use	Always
Assumptions	After logging in the user will also grant permission for the application to access the camera and keyboard of the device.
Use Case Diagram	UCD-1

Use Case ID	UC-2 Create New Notebook
Created:	10/18/2017
Actors:	Application User
Description:	This use case will describe the path a user can take to create a new notebook.
Trigger:	The user taps the new notebook button.
Precondition:	PRE-1: The user is authenticated in the application.
Postcondition:	A new notebook object is rendered on the screen on the chosen plane anchor.
Normal Flow:	1. Follow either flow for UC-1. 2. The user scans the room to allow ARKit to gather room information. 3. The user chooses the new notebook button. 4. The user chooses a notebook cover style. 5. The user chooses a flat surface that highlighted by a focus square in the view. 6. The 3D notebook model will be placed on that flat surface.
Alternative Flow:	1. Follow either flow for UC-1. 2. The user scans the room to allow ARKit to gather room information. 3. The user places a previous notebook. UC-3). 4. The user taps the new notebook button. 5. The user chooses a page cover style. 6. The previously placed notebook is removed and replaced with the new notebook.
Frequency of Use:	Frequently
Assumptions:	There is a flat surface detected as a plane inside the view.
Use Case Diagram:	UCD-1

Use Case ID	UC-3 Retrieve Previous Notebook
Created:	10/18/2017
Actors:	Application User
Description:	This use case will describe the path a user takes to access a previously added notebook.
Trigger:	The user taps the recent notebook button.
Precondition:	PRE-1, PRE-2: The user has a previously saved notebook.
Postcondition:	The chosen previous notebook is rendered on the screen.
Normal Flow:	1. Follow the process of UC-1 if necessary. 2. The user scans the room to detect a flat surface. 3. The user taps the chosen flat surface. 4. By default, the last notebook is placed. 5. The user taps the button to choose a different notebook. 6. Choose a previously created notebook from the list of notebooks.
Alternative Flow:	None
Frequency of Use:	Frequently
Assumptions:	None
Use Case Diagram:	UCD-2

Use Case ID	UC-4 Add A New Notebook Page
Created:	10/18/2017
Actors:	Application User
Description:	This use case describes the path a user can take to add a new notebook page.
Trigger:	The user taps add a new notebook page button.
Precondition:	PRE-3: There is a notebook open on the application.
Postcondition:	A new page is added to the notebook and auto saved to the user's account.
Normal Flow:	1. Follow the process of UC-1 if necessary. 2. Follow the process for UC-2 or UC-3. 3. Tap the button for adding a new page. 4. Choose a page template.
Alternative Flow:	The user can add new notebook pages in 3 situations: From the default recent notebook, from the path of adding a new blank notebook, or from the path of re-opening a previous notebook.
Frequency of Use:	Frequently
Assumptions:	None
Use Case Diagram:	UCD-3

Use Case ID	UC-5 Add Clipboard Text
Created:	10/18/2017
Actors:	Application User
Description:	This describes the path a user can take to paste clipboard text on a notebook page.
Trigger:	Pressing the clipboard button
Precondition:	PRE-3 The user has information on their keyboard in string format.
Postcondition:	The page will have the clipboard text in the available spot rendered on the screen.
Normal Flow:	1. Follow the process of UC-1 if necessary. 2. Follow the process for UC-2 or UC-3 . 3. Tap the button for clipboard text.
Alternative Flow:	None
Frequency of Use:	Frequently
Assumptions:	The clipboard text can be no larger than 140 characters.
Use Case Diagram:	UCD-5

Use Case ID	UC-6 Add Keyboard Text
Created:	10/18/2017
Actors:	Application User
Description:	This describes the path a user can take to input whatever text they want from the keyboard onto the notebook page.
Trigger:	The user completes typing on the screen
Precondition:	PRE-3 PRE-4: The current page has space available for new text.
Postcondition:	The keyboard input will be rendered on the open space on the page.
Normal Flow:	1. Follow the process of UC-1 if necessary. 2. Follow the process for UC-2 or UC-3. 3. Follow the process for UC-4 if necessary. 4. Tap the input text field and begin typing. 5. Tap off the keyboard to minimize the keyboard.
Alternative Flow:	1. Follow the process of UC-1 if necessary. 2. Follow the process for UC-2 or UC-3. 3. Tap the input text field and begin typing. 4. Tap off the Return button on the on-screen keyboard.
Alternative Flow 2:	1. Follow the process of UC-1 if necessary. 2. Follow the process for UC-2 or UC-3. 3. Tap the input text field and begin typing. 4. Tap on the Update text button.
Frequency of Use:	Frequently
Assumptions:	The input text will be less than 140 characters.
Use Case Diagram	UCD-9

Use Case ID	UC-7 Add an Image
Created:	10/18/2017
Actors:	Application User
Description:	This describes the path a user can take to choose an image from their phone's gallery and put it on a notebook page.
Trigger:	The user taps the add image button.
Precondition:	PRE-3, PRE-4
Postcondition:	The chosen gallery image is rendered on the page.
Normal Flow:	1. Follow the process of UC-1 if necessary. 2. Follow the process for UC-2 or UC-3 . 3. Choose to upload an image from the gallery to open the photo viewer. 4. Choose the image from the gallery you want to be added to the page. 5. The image will be the default template size.
Alternative Flow:	1. Follow the process of UC-1 if necessary. 2. Follow the process for UC-2 or UC-3. 3. Choose to upload an image from the gallery to open the photo viewer. 4. Choose the image from the gallery you want to be added to the page. 5. Drag on the edges of the image to resize the image within the notebook page.
Frequency of Use:	Frequently
Assumptions:	The user has pictures saved in their device's gallery.
Use Case Diagrams:	UDC-4

Use Case ID	UC-8 Delete Notebook
Created:	10/18/2017
Actors:	Application User
Description:	This describes the action a user needs to take to delete the entire notebook.
Trigger:	The user taps the button to delete the notebook.
Precondition:	There is notebook object on the screen.
Postcondition:	The notebook will be removed from the view and dropped from the database.
Normal Flow:	1. Follow for process for UC-1 if necessary. 2. If the default notebook is not desired Follow UC-3. 3. Tap the delete notebook button.
Alternative Flow:	None
Frequency of Use:	Rarely
Assumptions:	The user is the owner of the notebook being deleted.
Use Case Diagrams:	UDC-6

Use Case ID	UC-9 Delete a Notebook Page
Created:	10/18/2017
Actors:	Application User
Description:	This use case describes the process a user will follow to delete a page from the notebook.
Trigger:	Tapping the delete notebook button.
Precondition:	There is notebook object on the screen, with the current page showing being the one to delete.
Postcondition:	The notebook page that was shown is removed from the display and dropped from the database.
Normal Flow:	1. Follow UC-1 if necessary. 2. Follow UC-2 or UC-3. 3. Swipe to navigate to the desired page. 4. Tap the delete page button.
Alternative Flow:	None
Frequency of Use:	Rarely
Assumptions:	The user is the owner of the notebook.
Use Case Diagram:	UDC-7

Use Case ID	UC-10 Undo an Inserted Item
Created:	10/18/2017
Actors:	Application User
Description:	This describes the flow of a user to undo an insertion of a text or gallery item.
Trigger:	The user taps the undo button.
Precondition:	The user has inserted a text or gallery item in the current session.
Postcondition:	The last item that was inserted is removed from the notebook page.
Normal Flow:	1. Follow the process for UC-1 if necessary. 2. Follow the process for UC-2 or UC-3. 3. Follow the process for UC-6 Or UC-7. 4. Press the Undo button.
Alternative Flow:	None
Frequency of Use:	Frequently
Assumptions:	The user understands that the deletion of pages and notebooks cannot be undone.

Use Case ID	UC-11 Share A Notebook
Created:	10/18/2017
Actors:	Application User(s)
Description:	This describes the process a user must follow to share a notebook with another user.
Trigger:	The user taps the share button.
Precondition:	The user has a notebook created and has internet connection.
Postcondition:	The alternate application user has access to view the shared notebook on their device.
Normal Flow:	1. Follow UC-1 if necessary. 2. Follow process for UC-2 or UC-3. 3. Follow UC-6 or UC-7 if necessary 4. Tap the share button. 5. Retrieve the generated link for the notebook. 6. Message the notebook link to the user you wish to share the notebook with.
Alternative Flow:	None
Frequency of Use:	Rarely
Assumptions:	The user has cellular connection with the ability to send a text message to another application user.
Use Case Diagram:	UDC-8

Use Case ID	UC-12 Open a Shared Notebook
Created:	10/18/2017
Actors:	Application User(s)
Description:	This describes the process a user must follow to open a notebook that has been shared with them.
Trigger:	The user receives a message with a dynamic link to the shared notebook.
Precondition:	PRE-1
Postcondition:	The application user has access to view the shared notebook on their device.
Normal Flow:	1. The user receives a text message with the link to the notebook content. 2. The user taps the link within the message to open the AR Notebook application. 3. Follow process for UC-1 if necessary. 4. Follow processes for placing a notebook in UC-1 or UC-2.
Alternative Flow:	None
Frequency of Use:	Frequently
Assumptions:	The user has cellular connection with the ability to receive a text message to another application user.
Use Case Diagram:	

Use Case ID	UC-13 Geocaching With Notebooks
Created:	10/28/2017
Actors:	Application User(s)
Description:	This describes the process a user must follow to open a notebook that has been stored at particular geographical location.
Trigger:	The user is in the vicinity where the notebook's GPS coordinates are located.
Precondition:	PRE-1
Postcondition:	The application user has access to view the geocached notebook on their device.
Normal Flow:	1. The user physically moves to the coordinates where a notebook is located. 2. An alert shall appear on the user's screen advising them of a notebook in their vicinity. 3. The user shall choose whether to retrieve said notebook. 4. The user shall follow UC-1 and/or UC-2 to place the retrieved notebook onto a flat surface.
Alternative Flow:	None
Frequency of Use:	Rarely
Assumptions:	The user has a stable cellular data connection so that they can view real time GPS data as well as notebook location data.

Use Case ID	UC-14 Alert Low Battery
Created:	10/28/2017
Actors:	Application User(s)
Description:	This describes the process a user must follow to use the application when below the minimum battery level, which is 5%.
Trigger:	The user's iOS device is below 5% battery.
Precondition:	None
Postcondition:	The user has access to the application or exits the application depending on their choice.
Normal Flow:	1. The user opens the application or is already using the application. 2. There will be an alert shown to the user that they must have at minimum 5% battery to use the application. 3. The user then must plug the device into a charging station. 4. The user taps the option stating "I am charging my phone" 5. Once verified that the user is above 5% battery or is currently charging the device, the user shall be able to access the application as normal.
Alternative Flow:	1. The user opens the application or is already using the application. 2. There will be an alert shown to the user that they must have at minimum 5% battery to use the application. 3. The user taps the option that exits the application. 4. The application closes.
Frequency of Use:	Rarely
Assumptions:	The user has access to a charging station if they wish to continue using the application.
Use Case Diagram:	UCD-10

Use Case ID	UC-15 User Has No Internet Connection
Created:	10/31/2017
Actors:	Application User(s)
Description:	This describes the process of a user's access inside the application after they lose internet connection.
Trigger:	The user tries to share, retrieve, or Geocache their notebooks.
Precondition:	The user is signed into the application with a Facebook account. The user then loses internet connection
Postcondition:	The user has access to any notebook functionality except retrieve, share, or Geocaching.
Normal Flow:	1. The user creates a notebook. 2. The inserts any piece of content. 3. The user tries to share the notebook. 4. There will be an alert that displays to the user that they need internet connection to use that feature.
Alternative Flow:	1. The user is at the main View Controller screen. 2. The user tries to retrieve a previous notebook under the same Facebook account 3. There will be an alert that displays to the user that they need internet connection to use that feature.
Frequency of Use:	Rarely
Assumptions:	The user was logged in with a valid Facebook account before losing internet connection.

4.3 Use Case Diagrams

UCD-1 Create Notebook

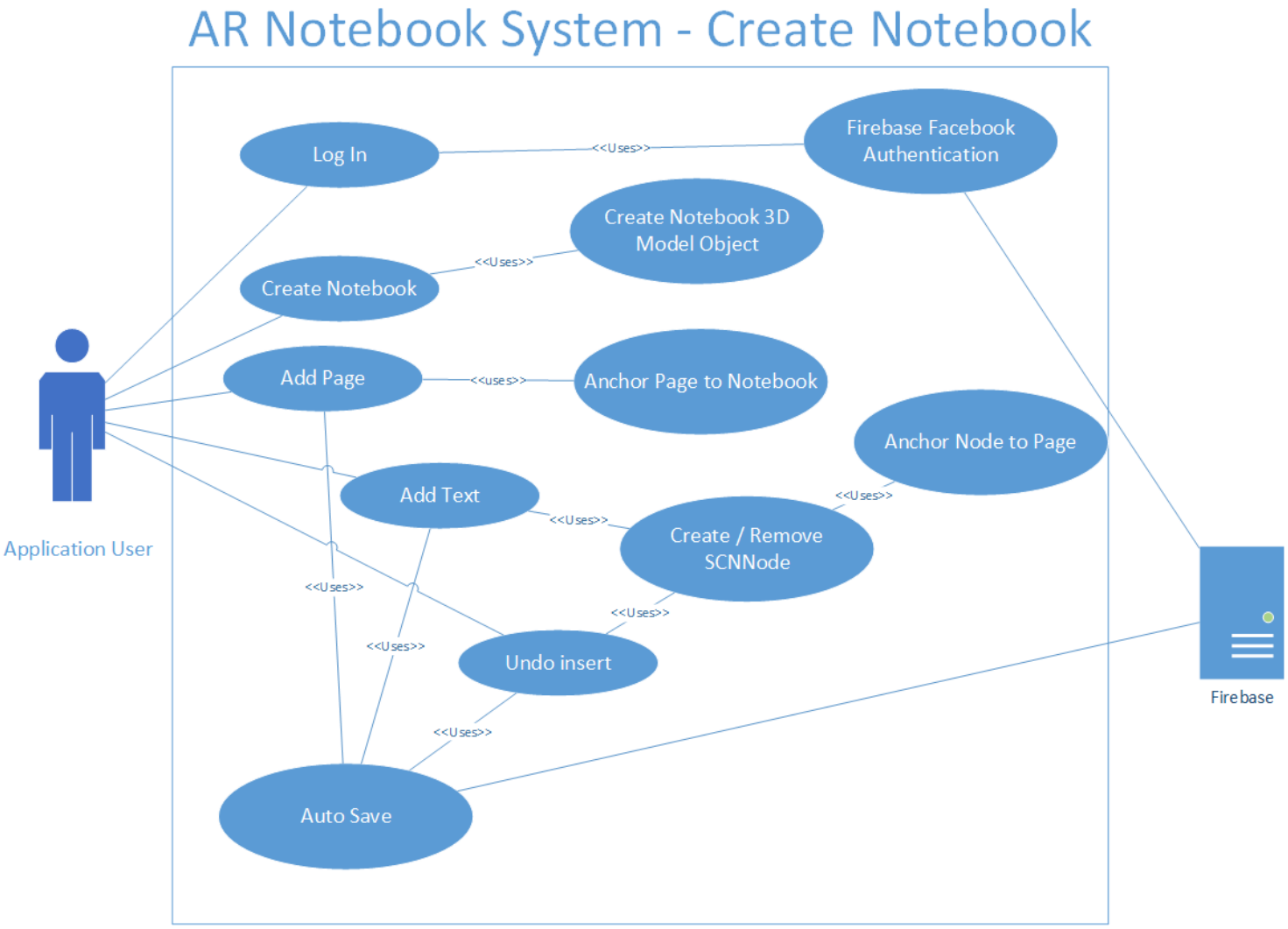


Figure 4 Create Notebook UCD

UCD-2 Open Recent Notebook

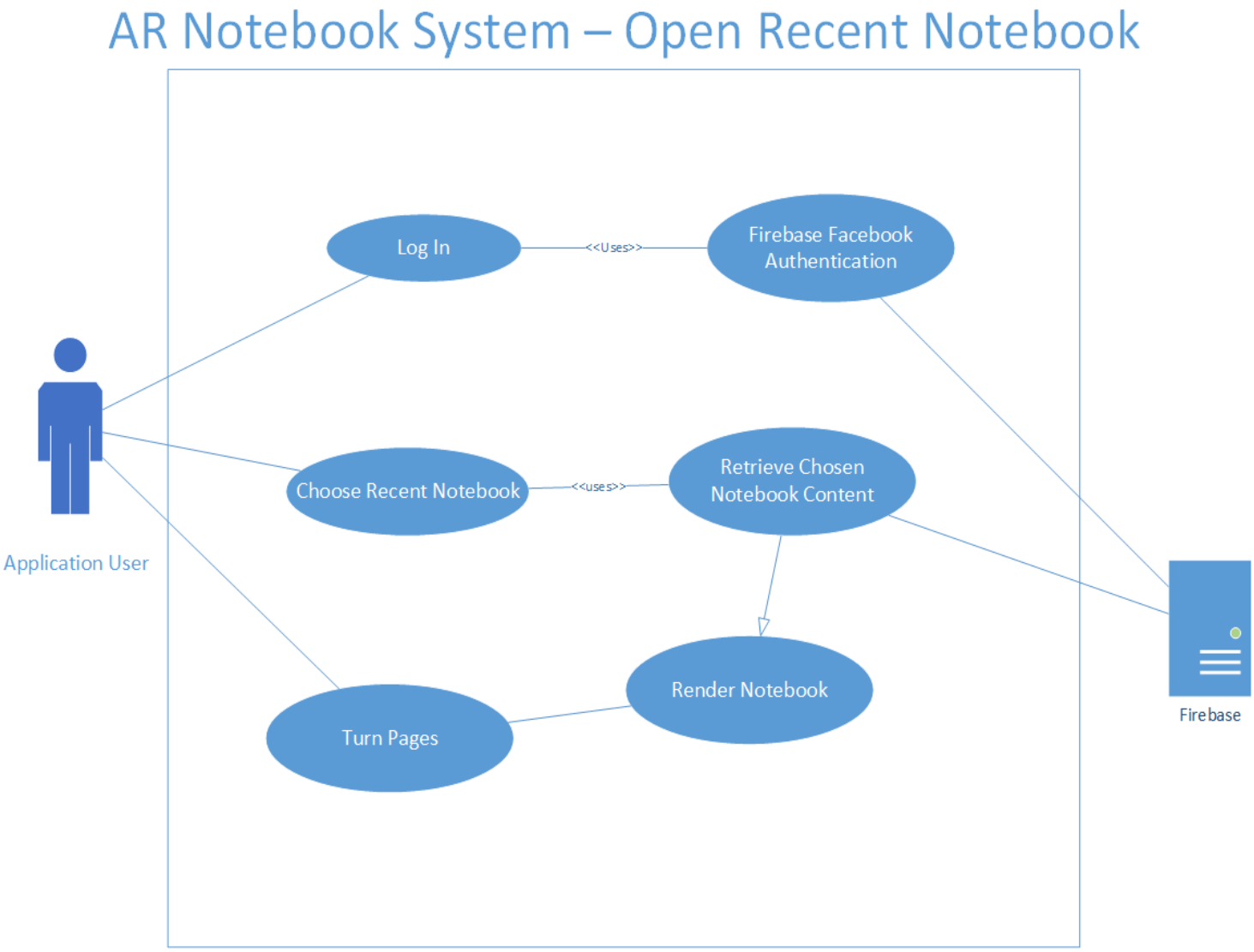


Figure 5 Recent Notebook UCD

UCD-3 Add a Page

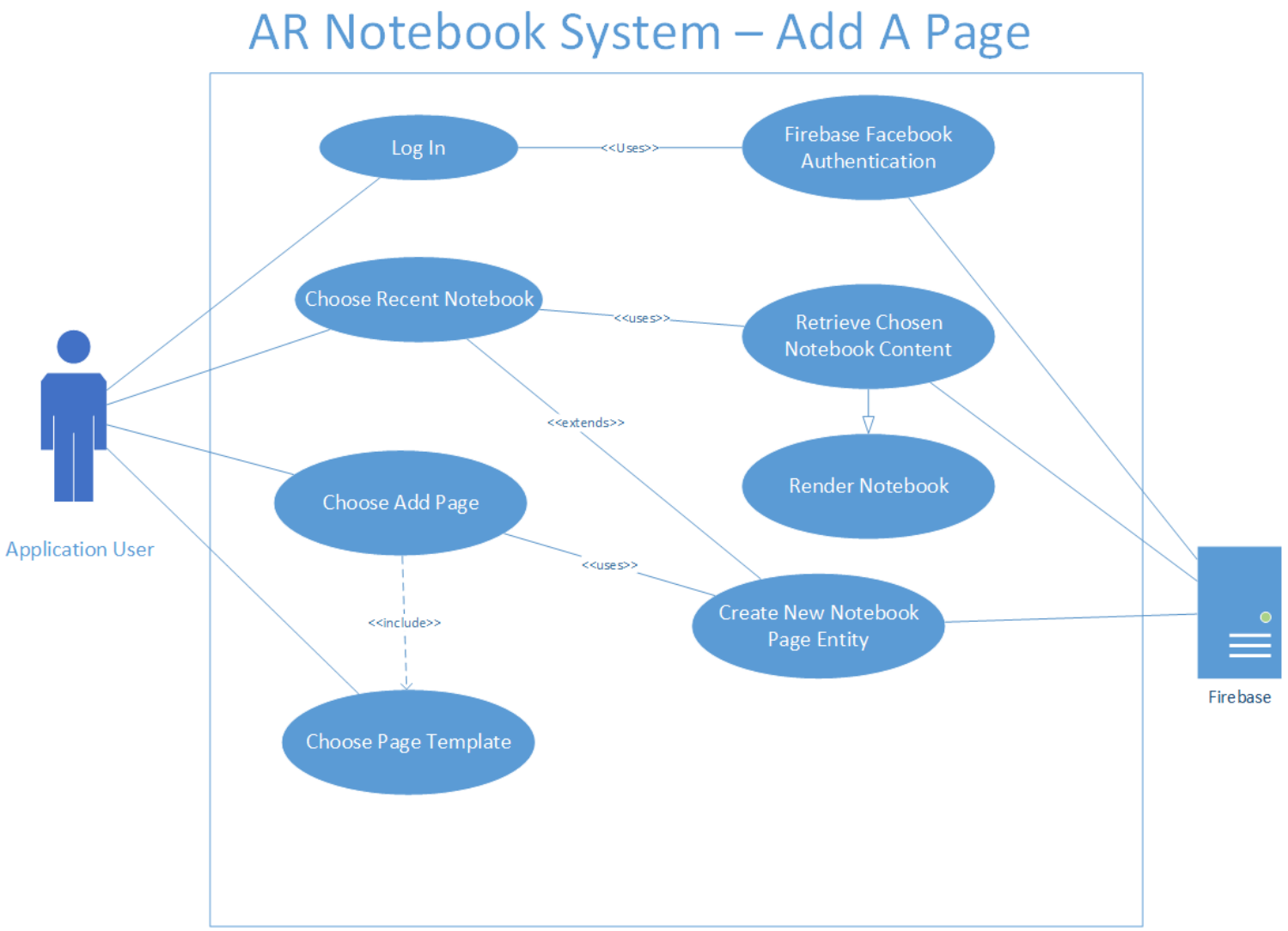


Figure 6 Add Page UCD

UCD-4 Insert Gallery

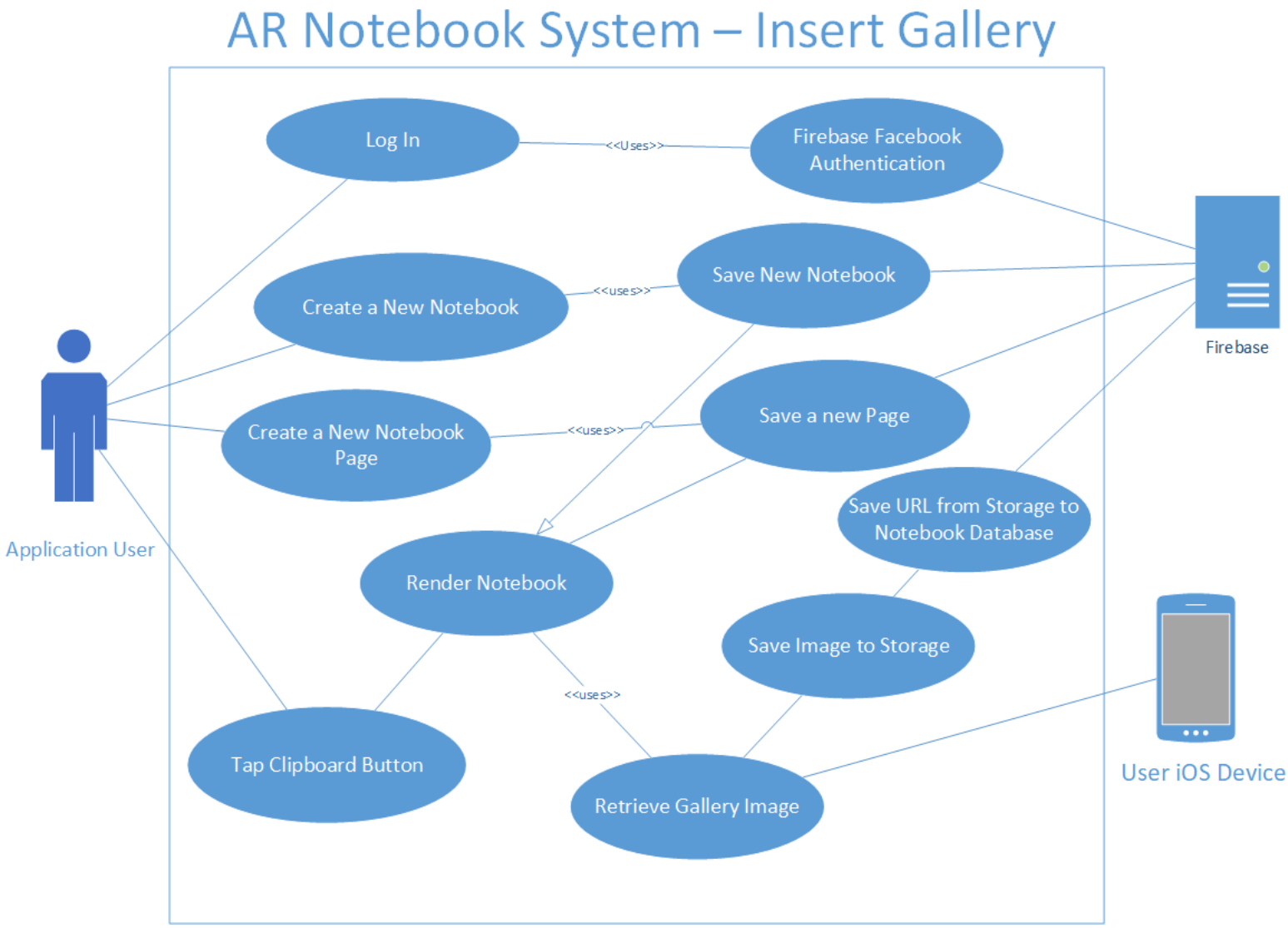


Figure 7 Insert Gallery UCD

UCD-5 Insert Clipboard

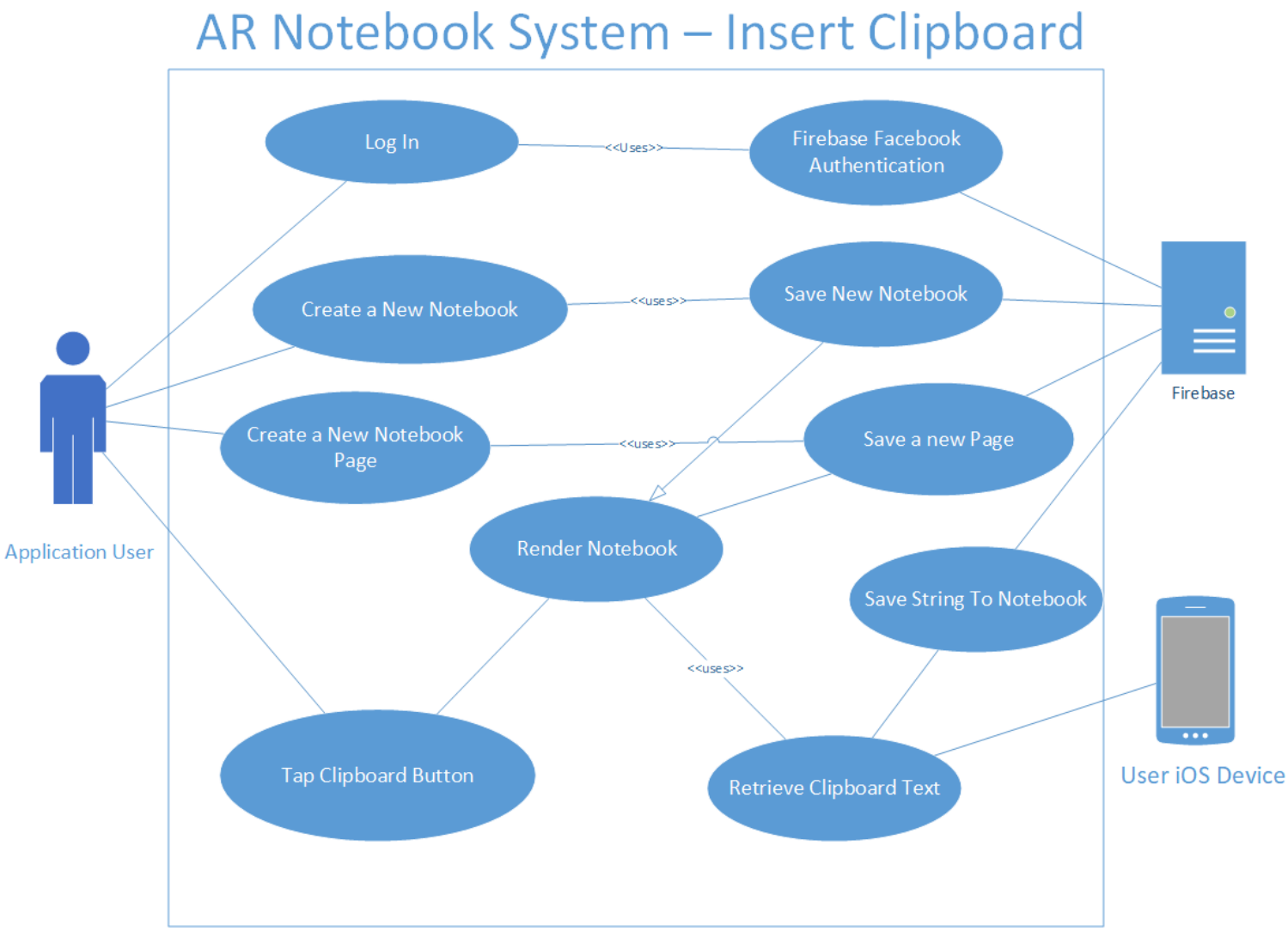


Figure 8 Insert Clipboard UCD

UCD-6 Delete a Notebook

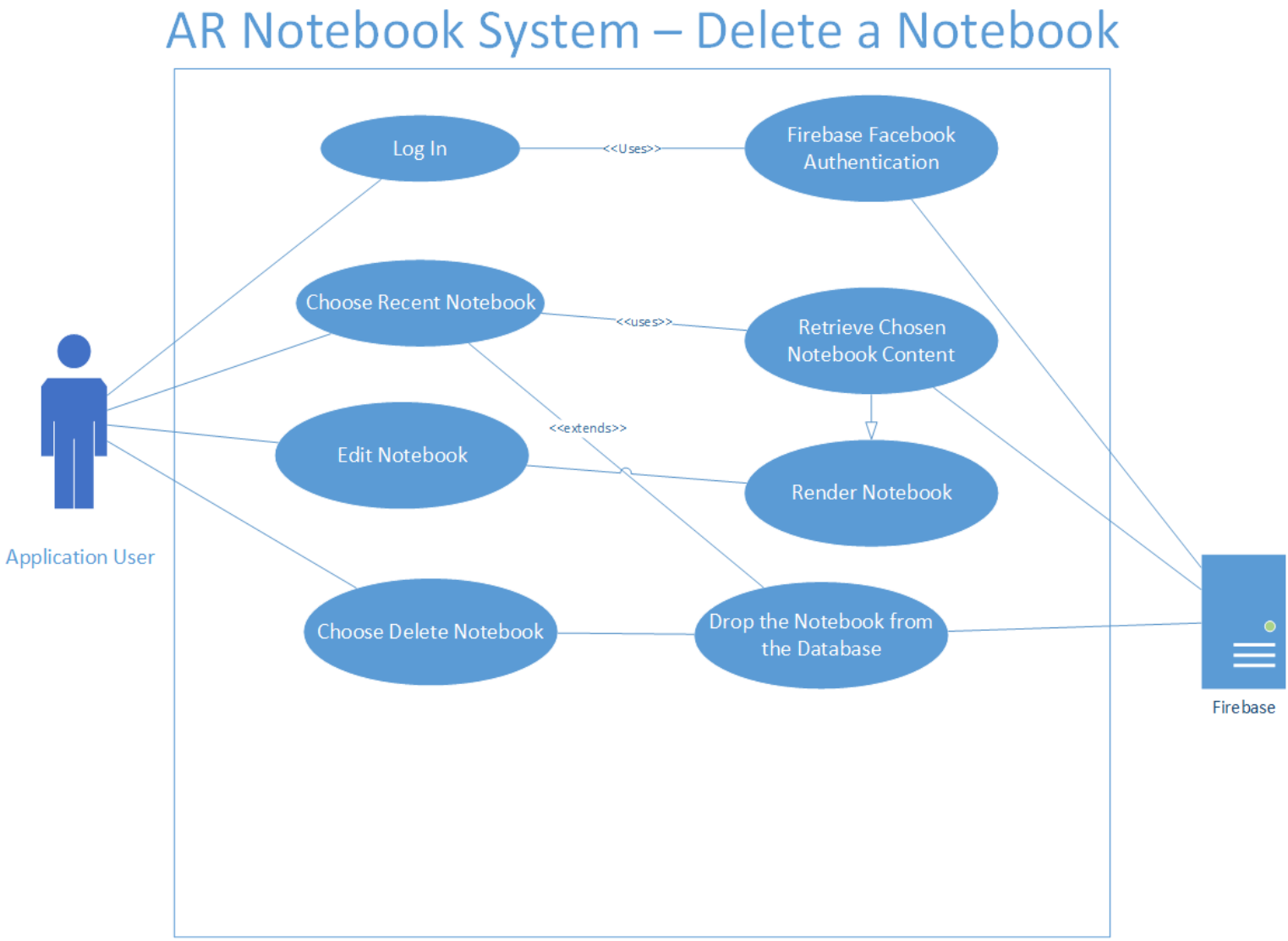


Figure 9 Delete Notebook UCD

UCD-7 Delete a Notebook Page

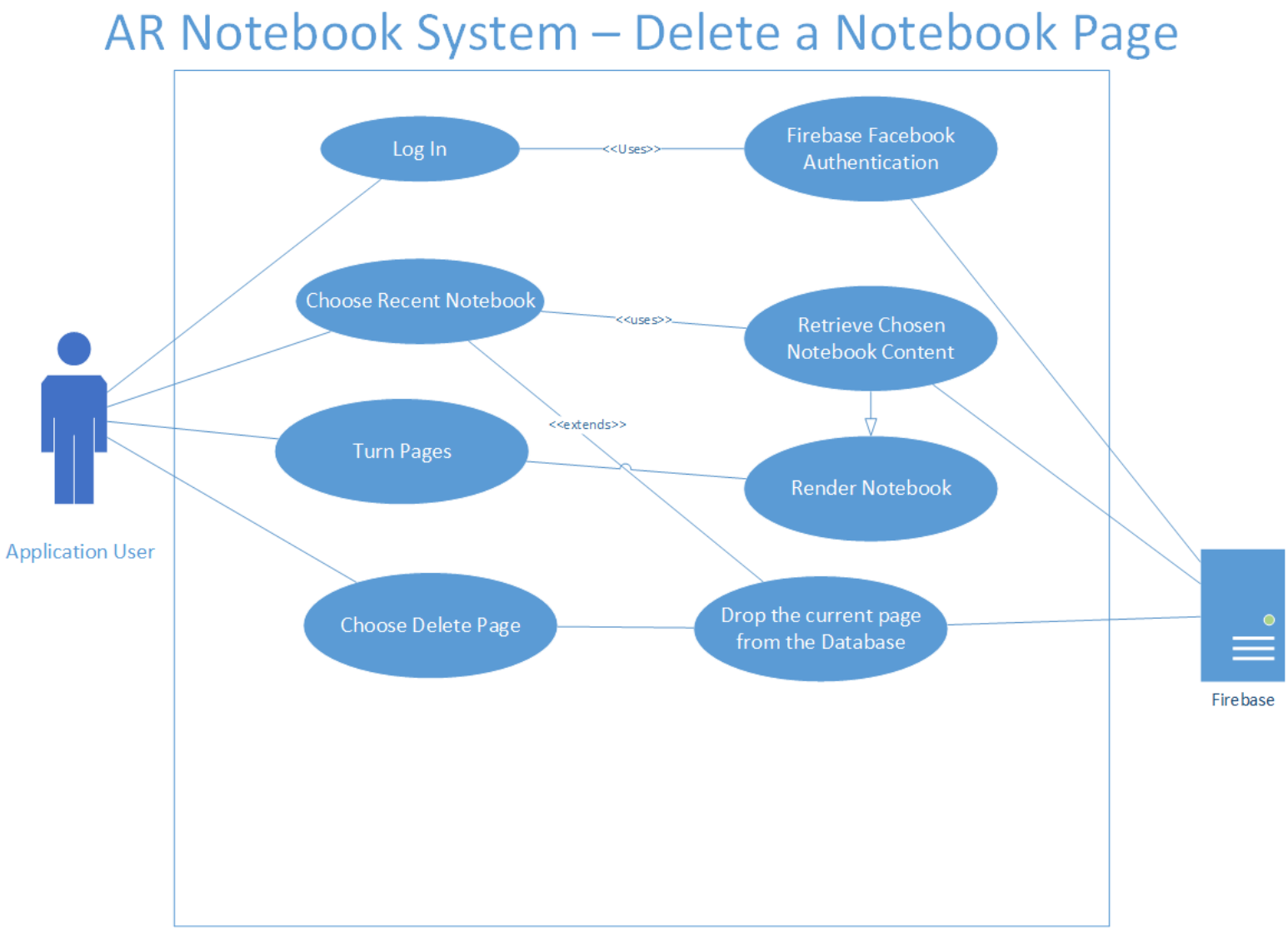


Figure 10 Delete Notebook Page UCD

UCD-8 Sharing

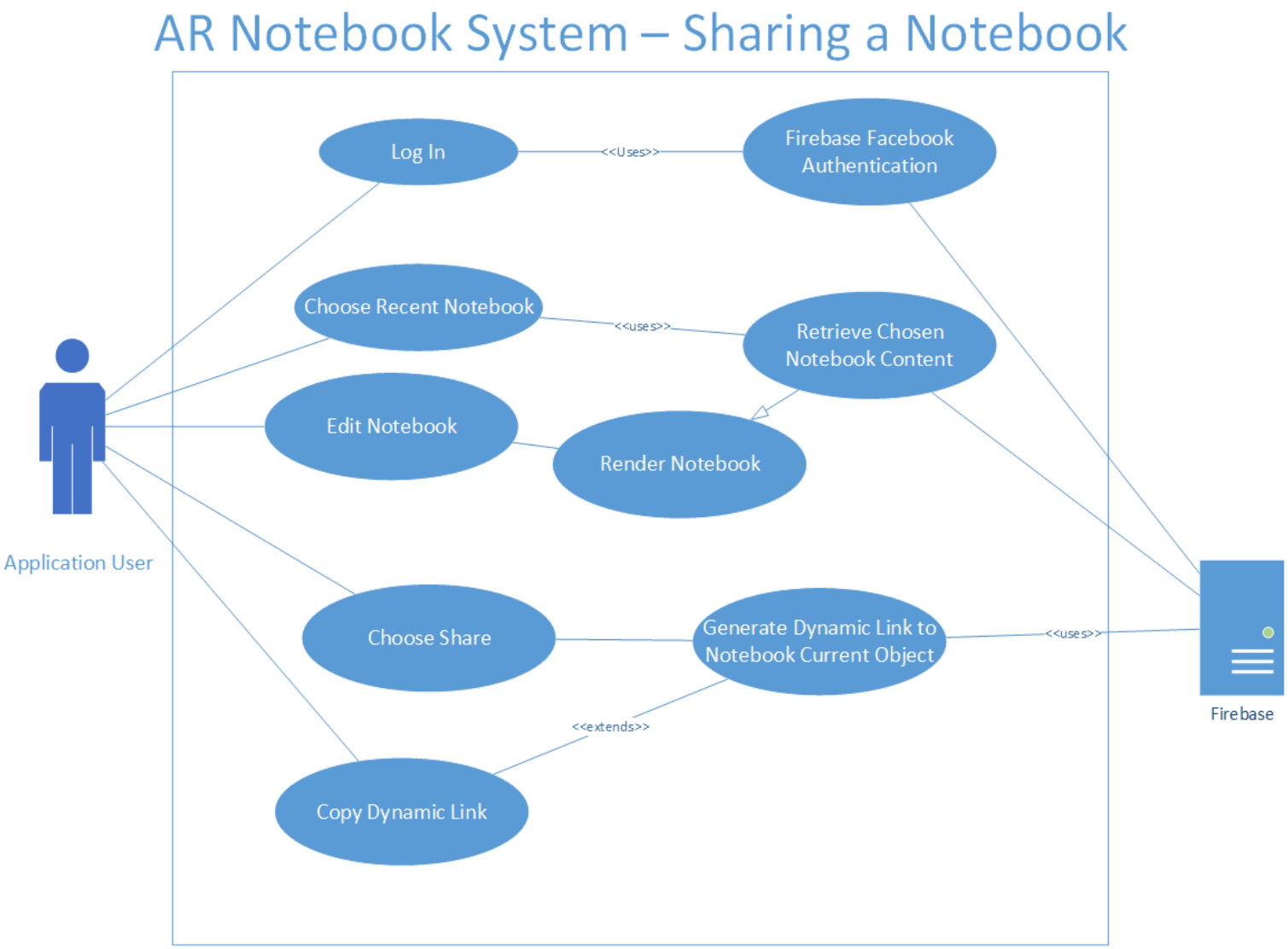


Figure 11 Share Notebook UCD

UCD-9 Insert Text

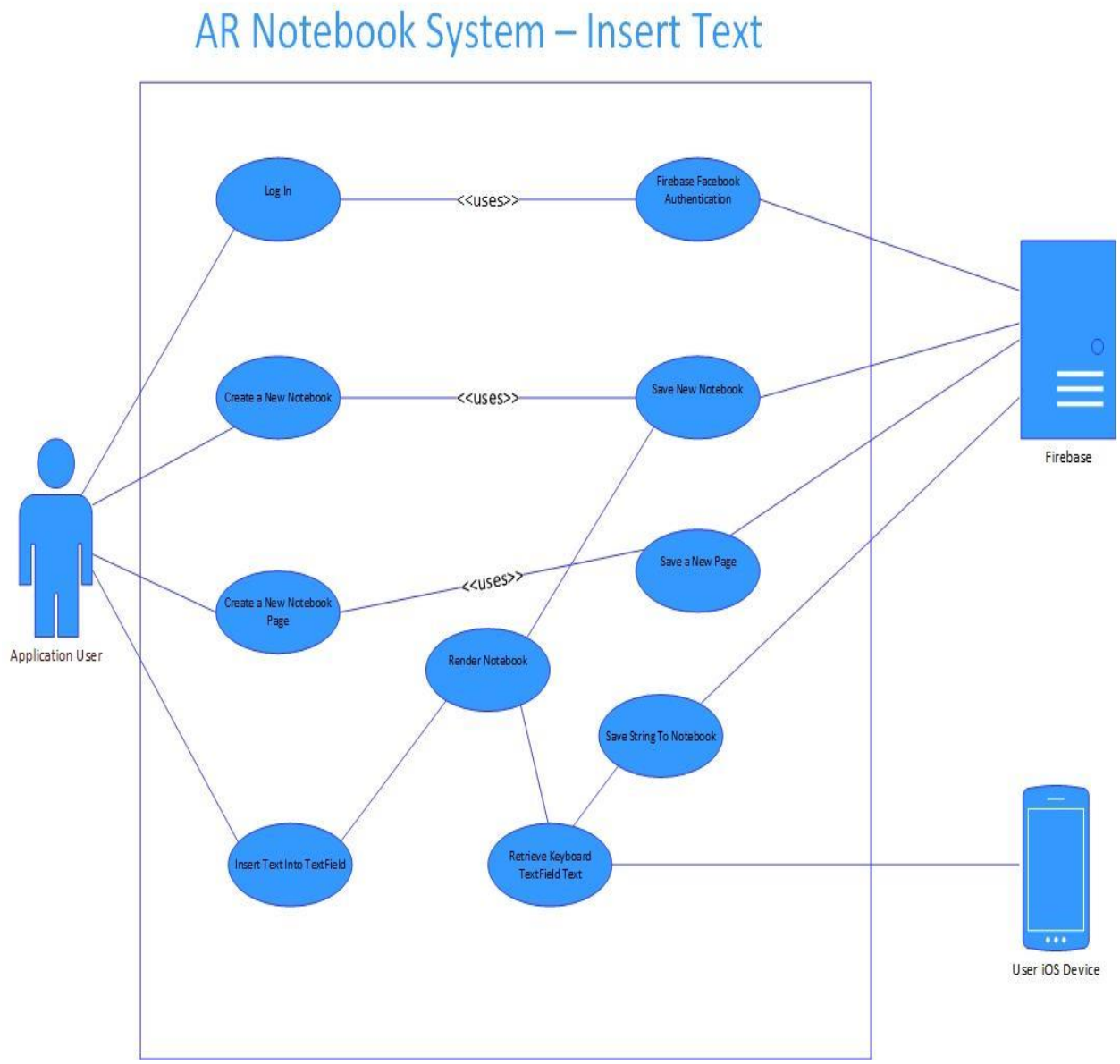


Figure 12 Insert Text UCD

UCD-10 Low Battery Alert

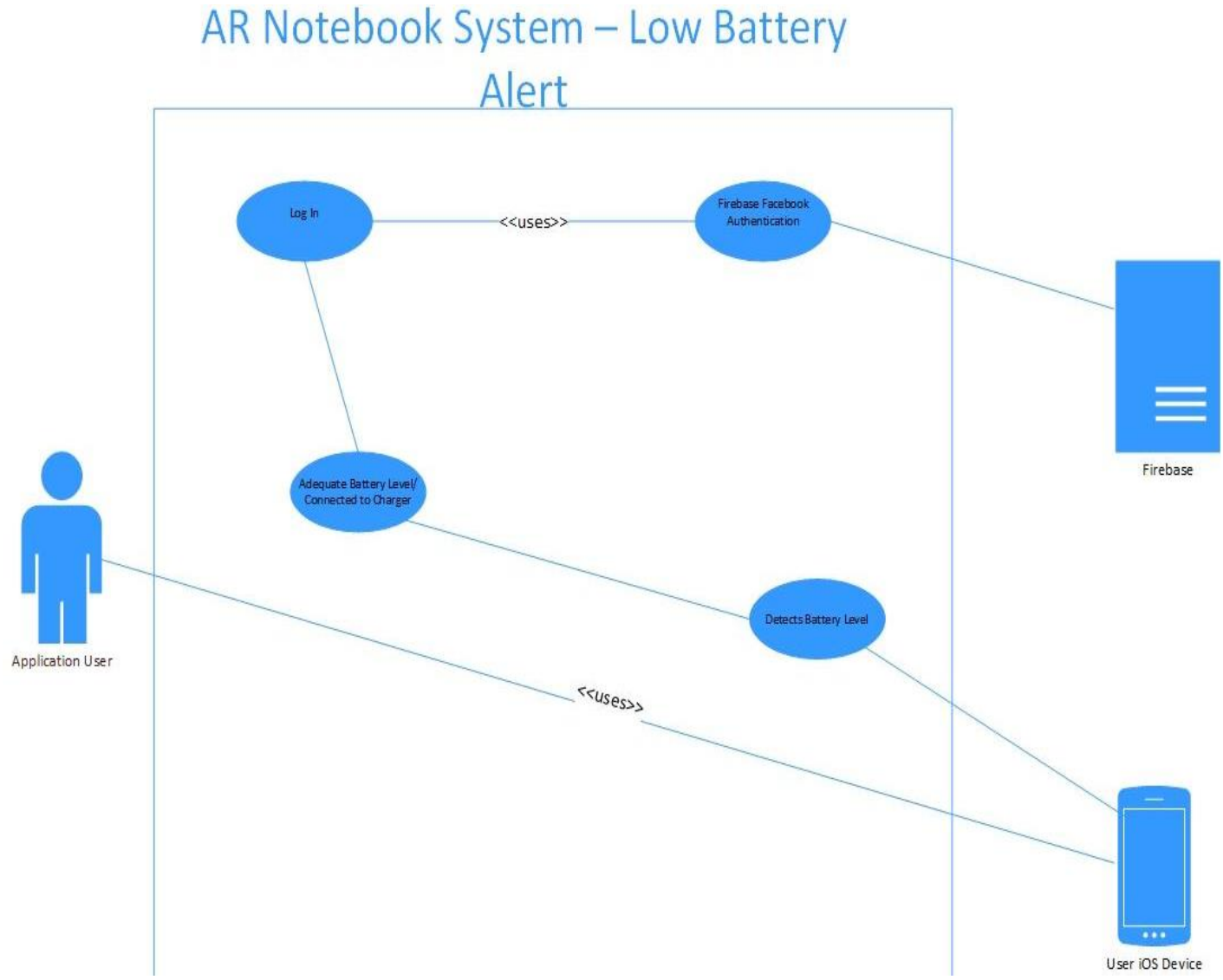


Figure 13Low Battery UCD

4.4 Sequence Diagram

4.4.1 Log In

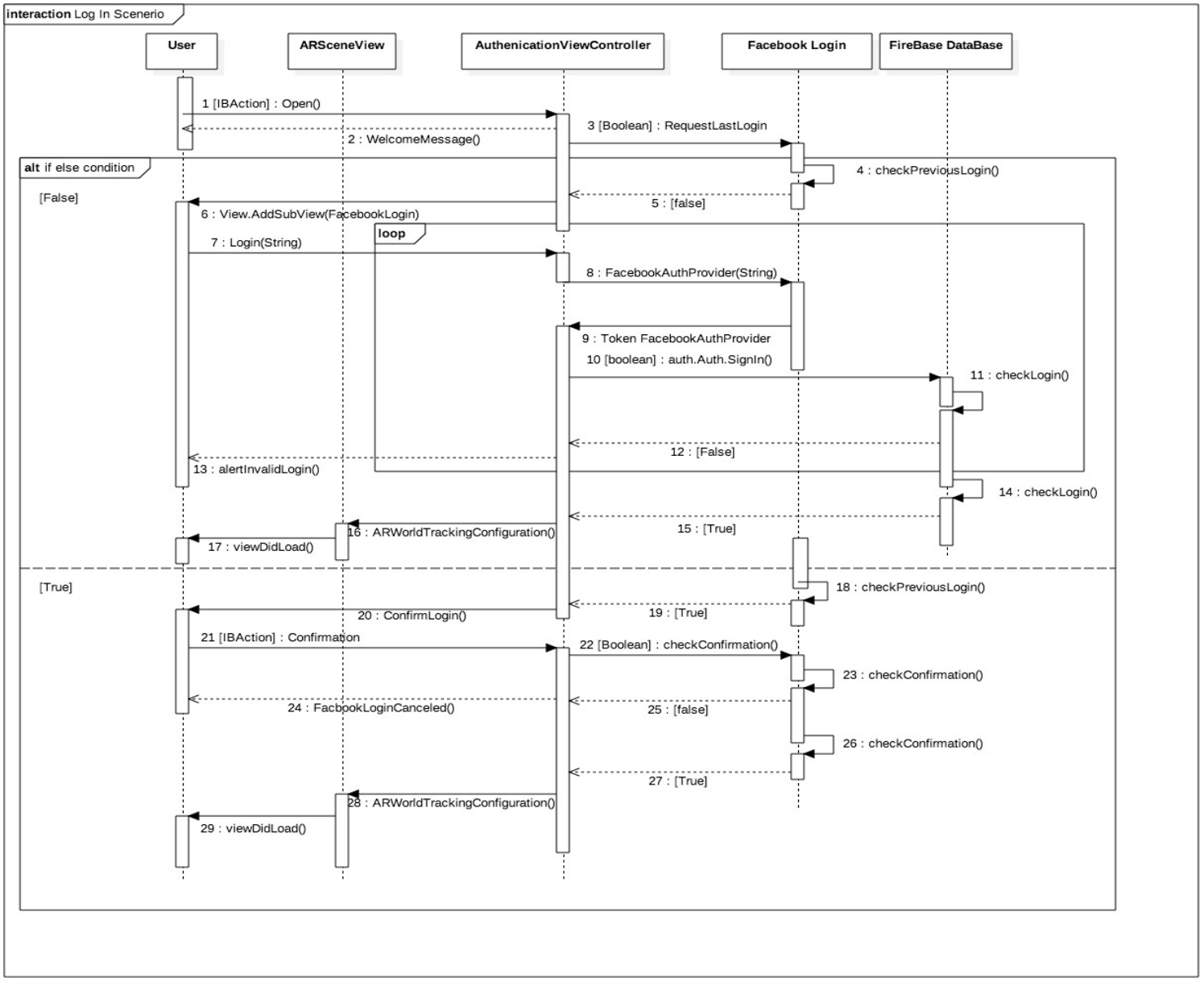


Figure 14 Log In SD

4.4.2 Plane Detection

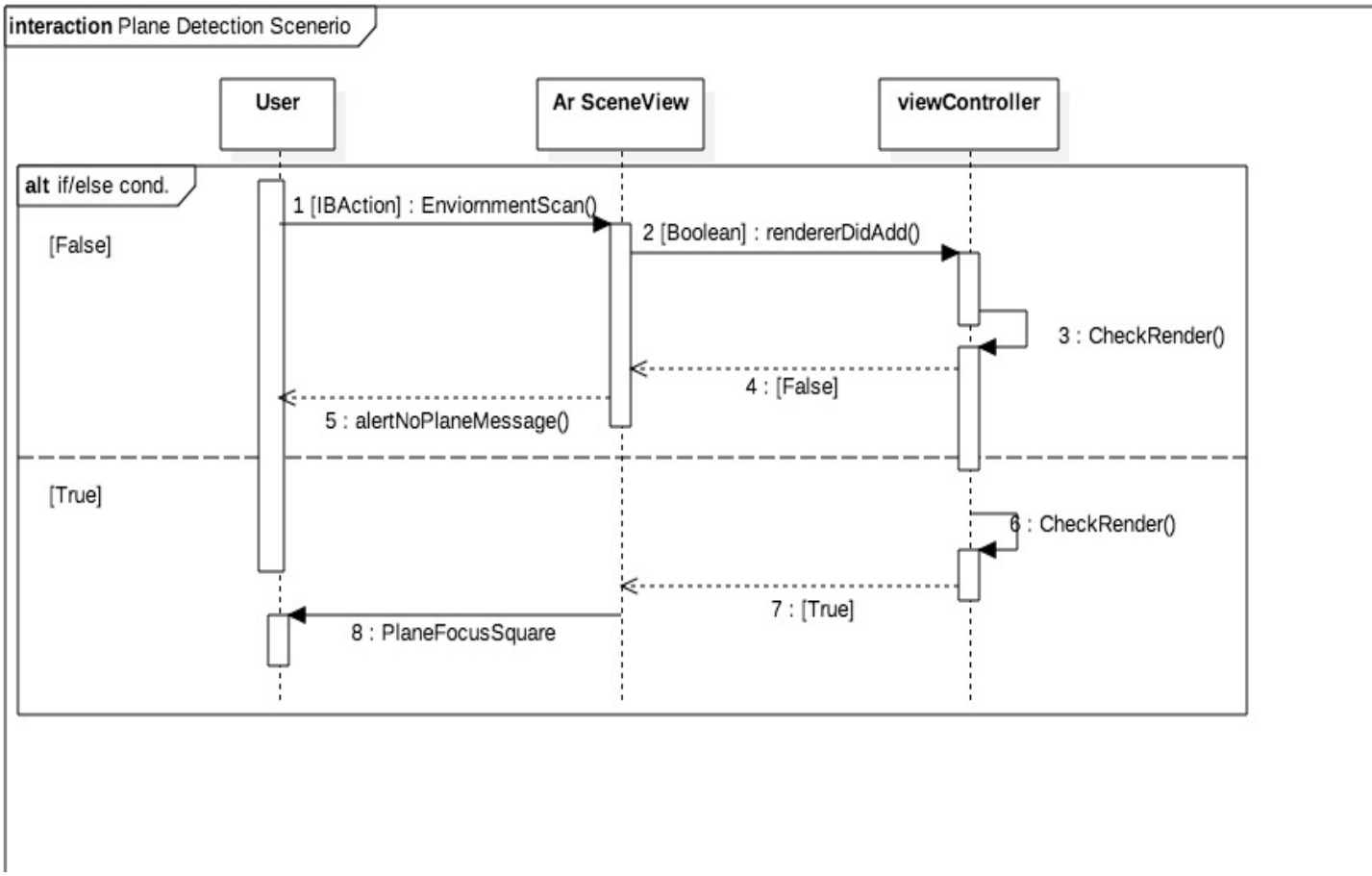


Figure 15 Plane Detection SD

4.4.3 Add a Notebook

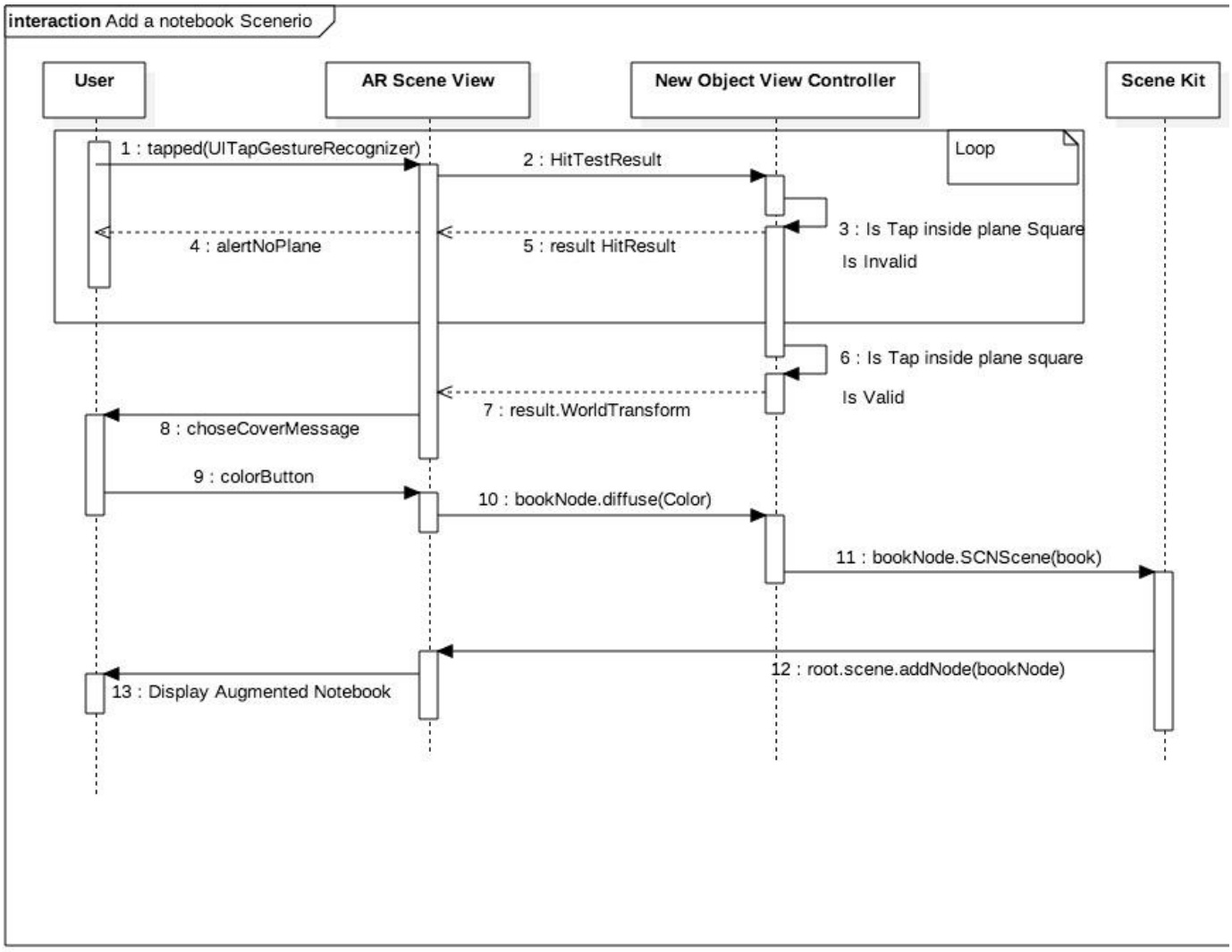


Figure 16 Add Notebook SD

4.4.4 Add a Page

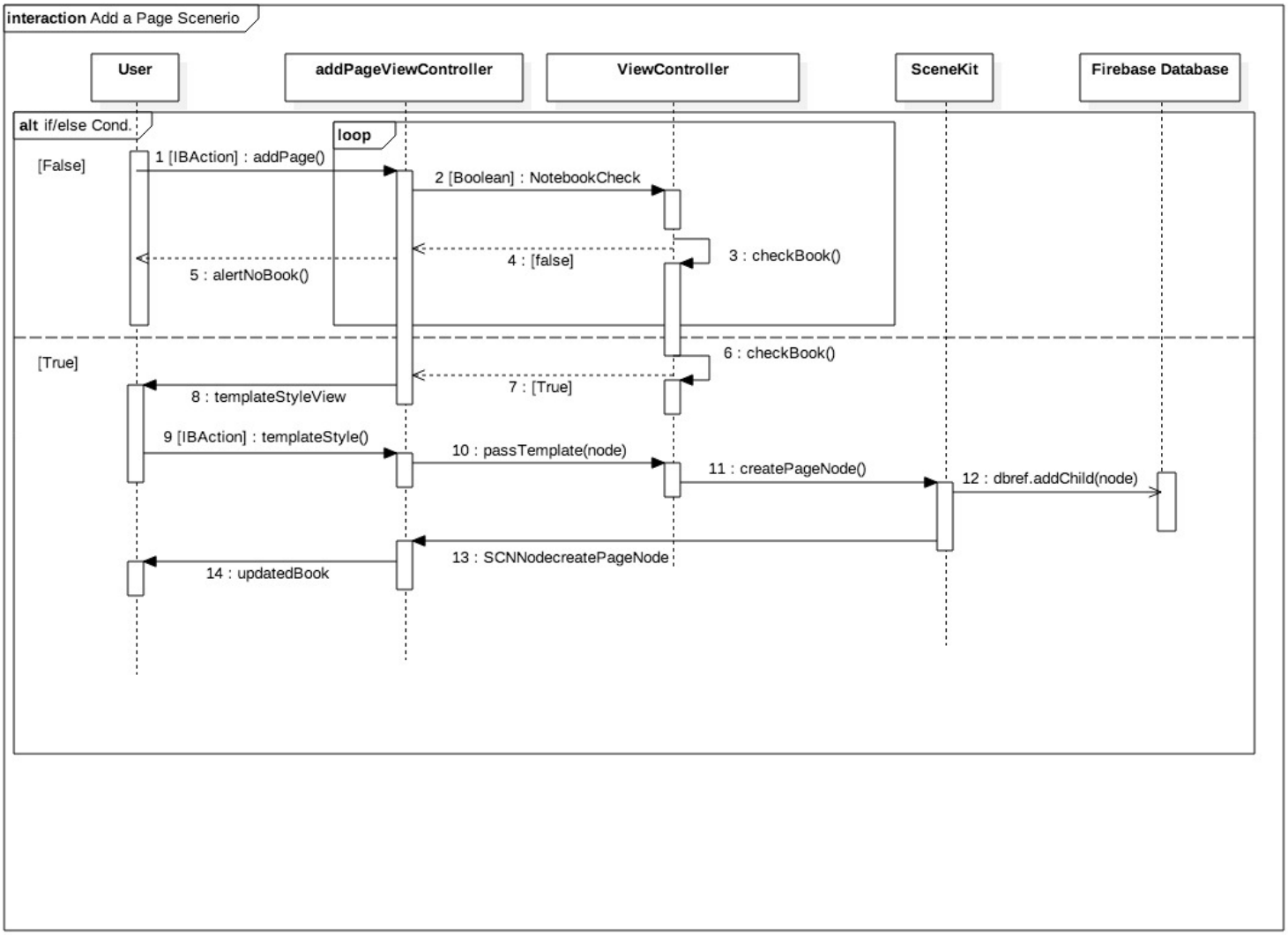


Figure 17 Add Page SD

4.4.5 Add Clipboard text

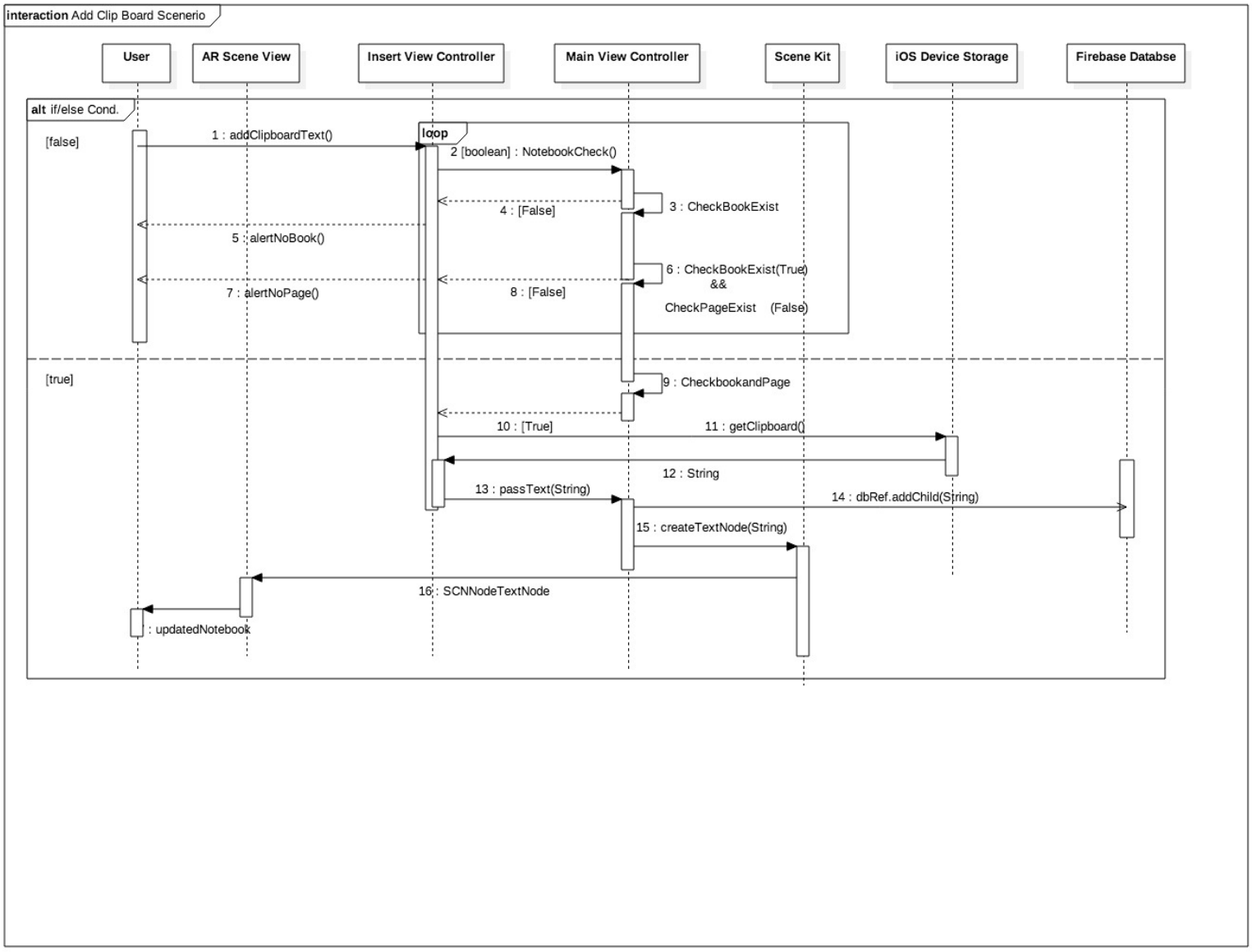


Figure 18 Add Clipboard SD

4.4.6 Delete Notebook

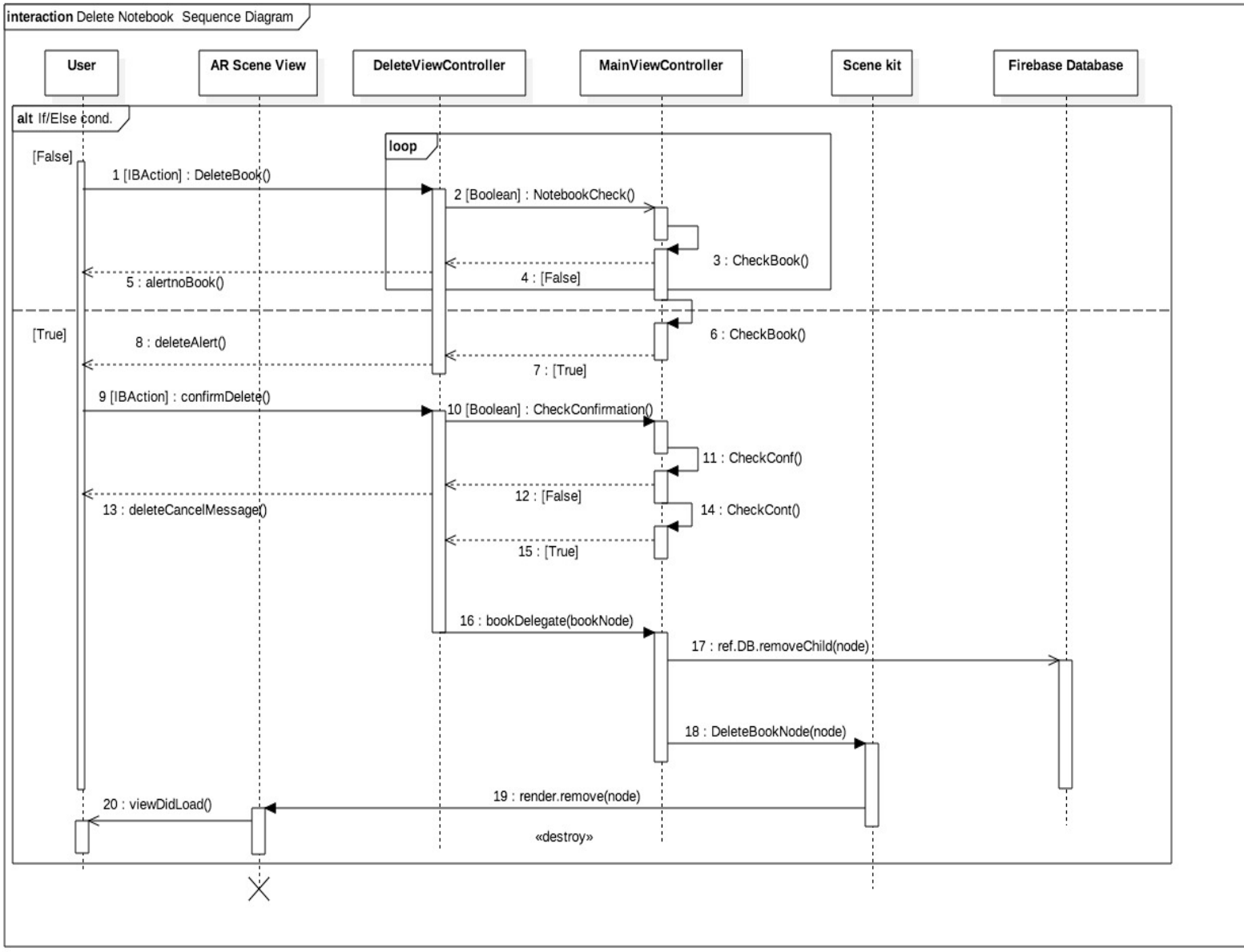


Figure 19 Delete Notebook SD

4.4.7 Delete Page

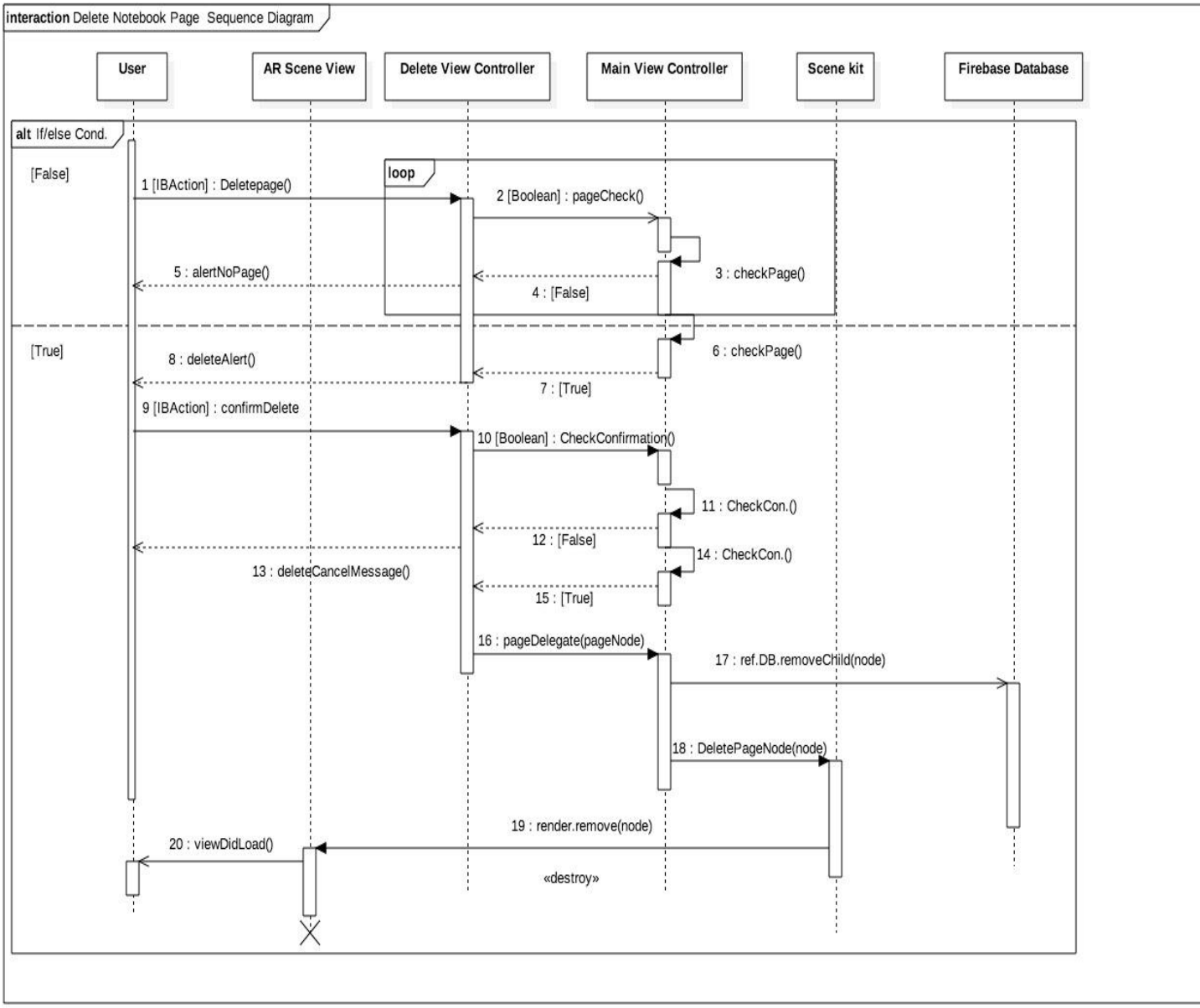


Figure 20 Delete Page SD

4.4.8 Undo

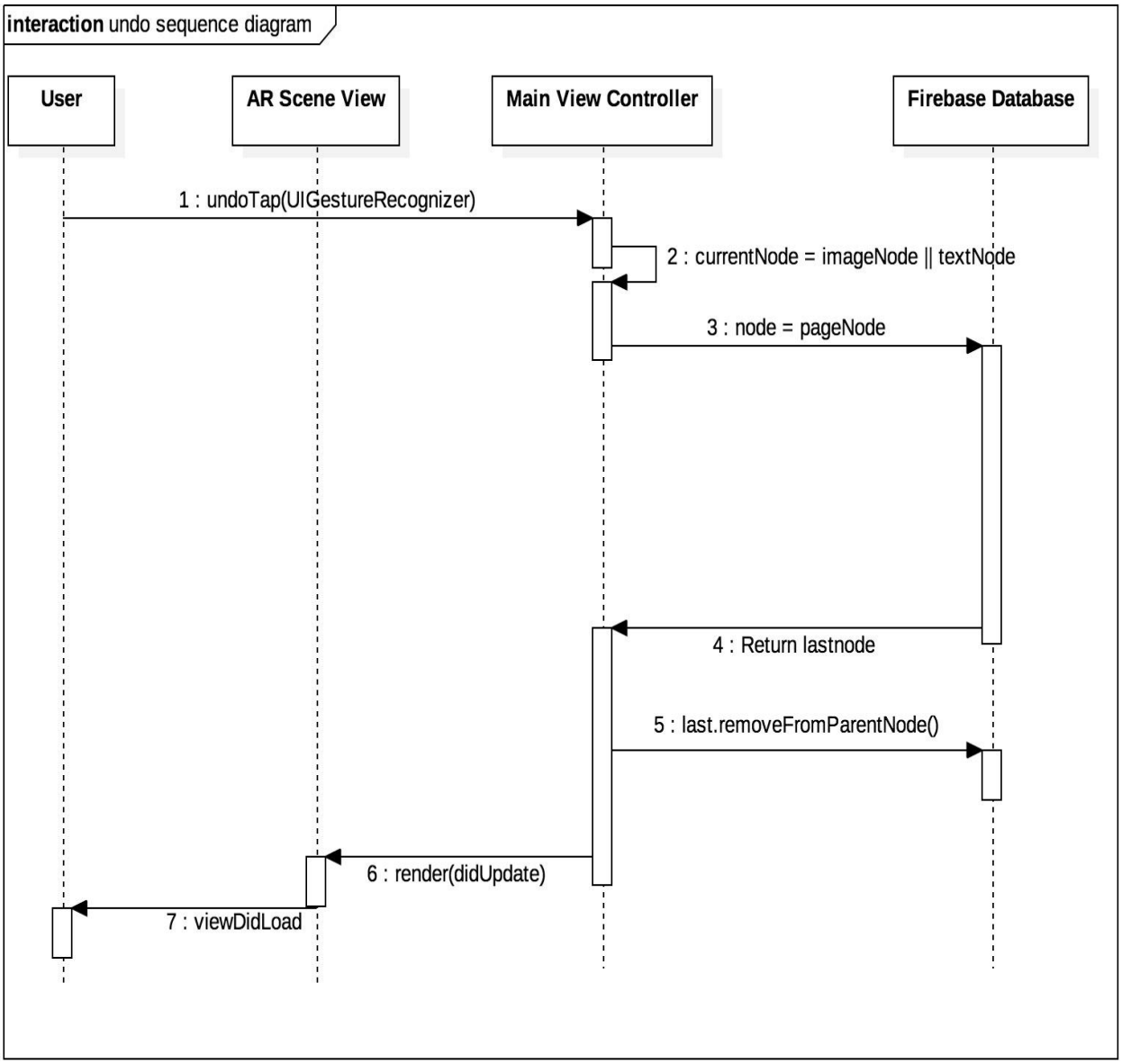


Figure 21 Undo SD

4.4.9 Insert Text

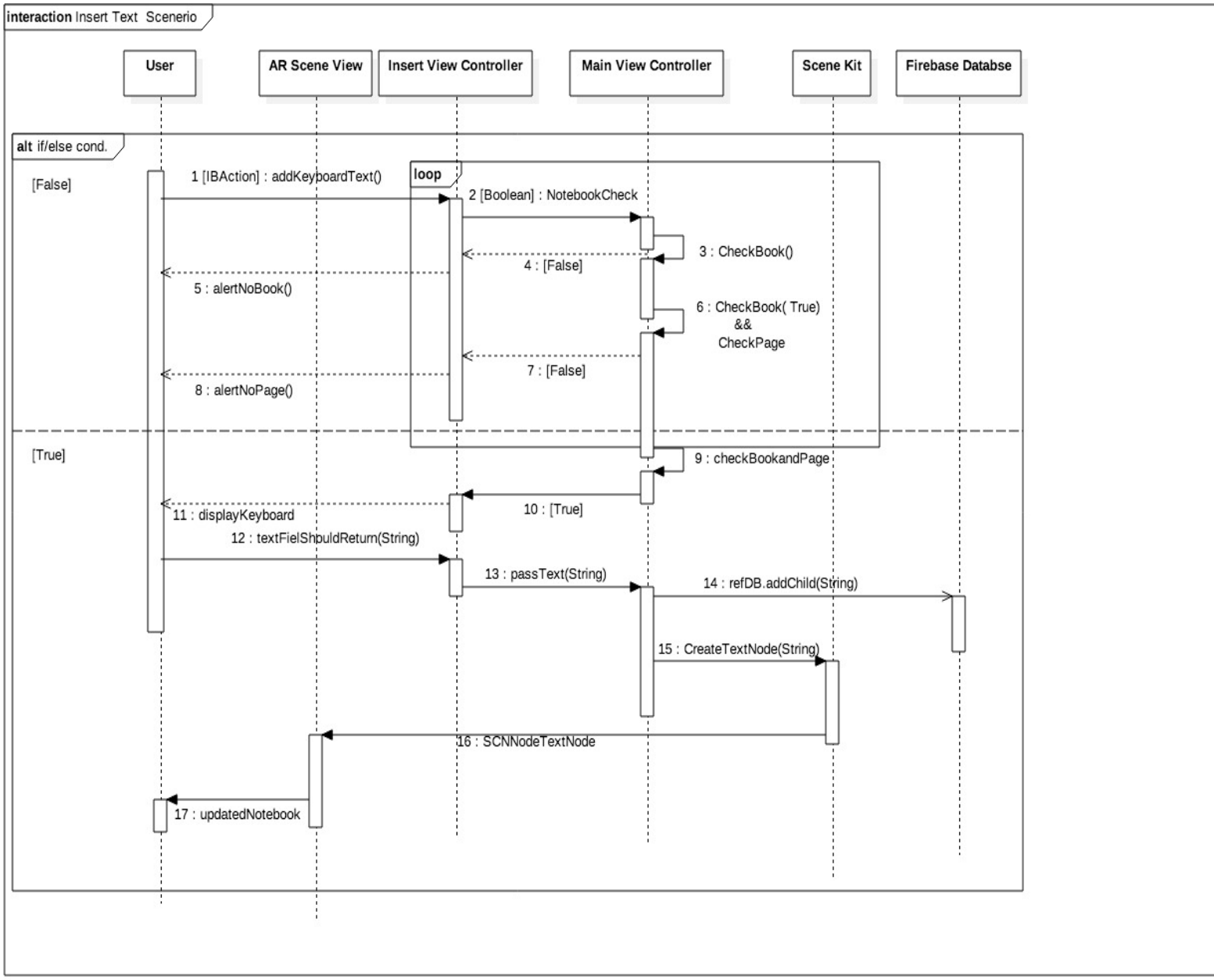


Figure 22 Insert Text SD

4.4.10 Insert Gallery

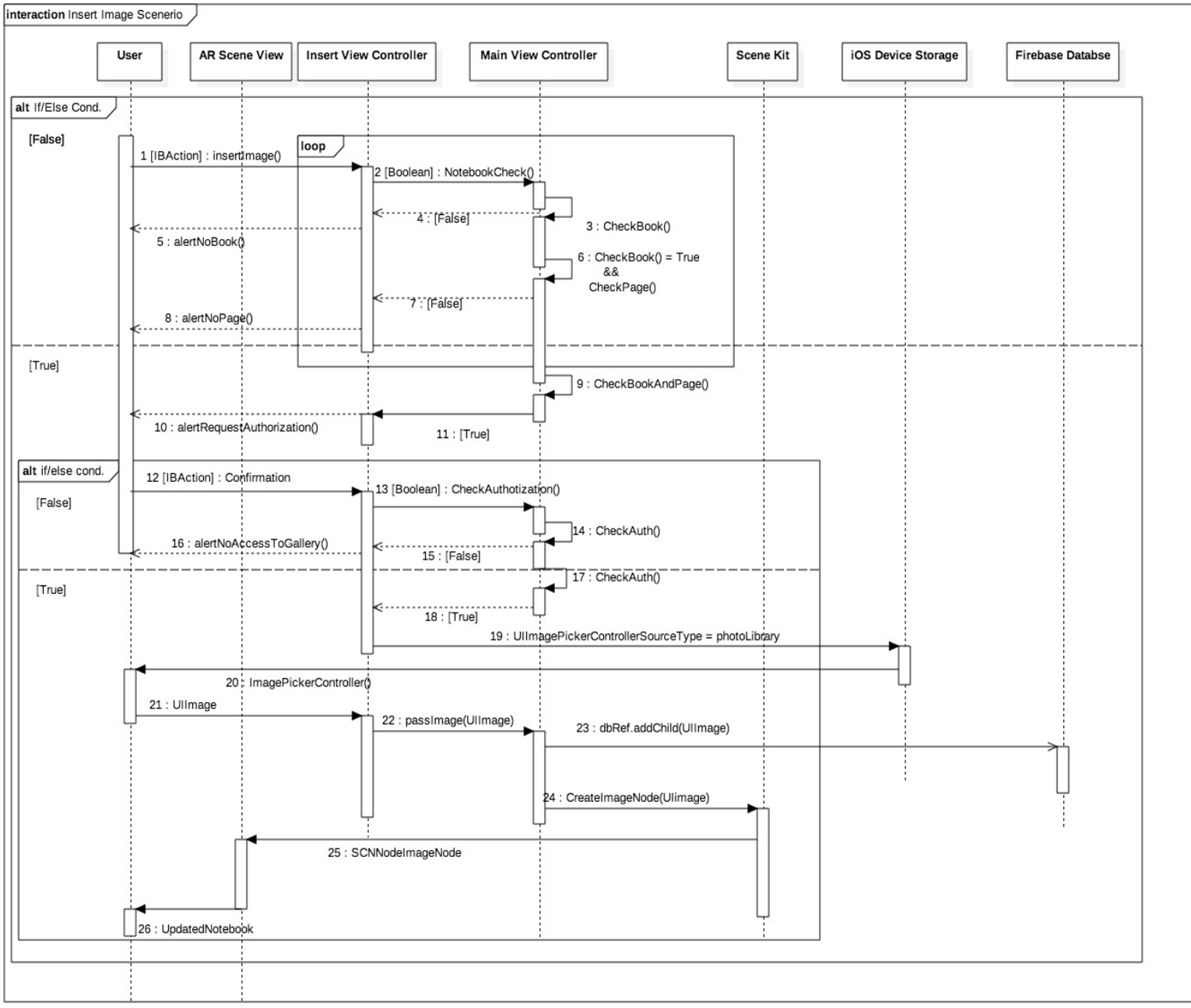


Figure 23 Insert Gallery SD

4.4.11 Share

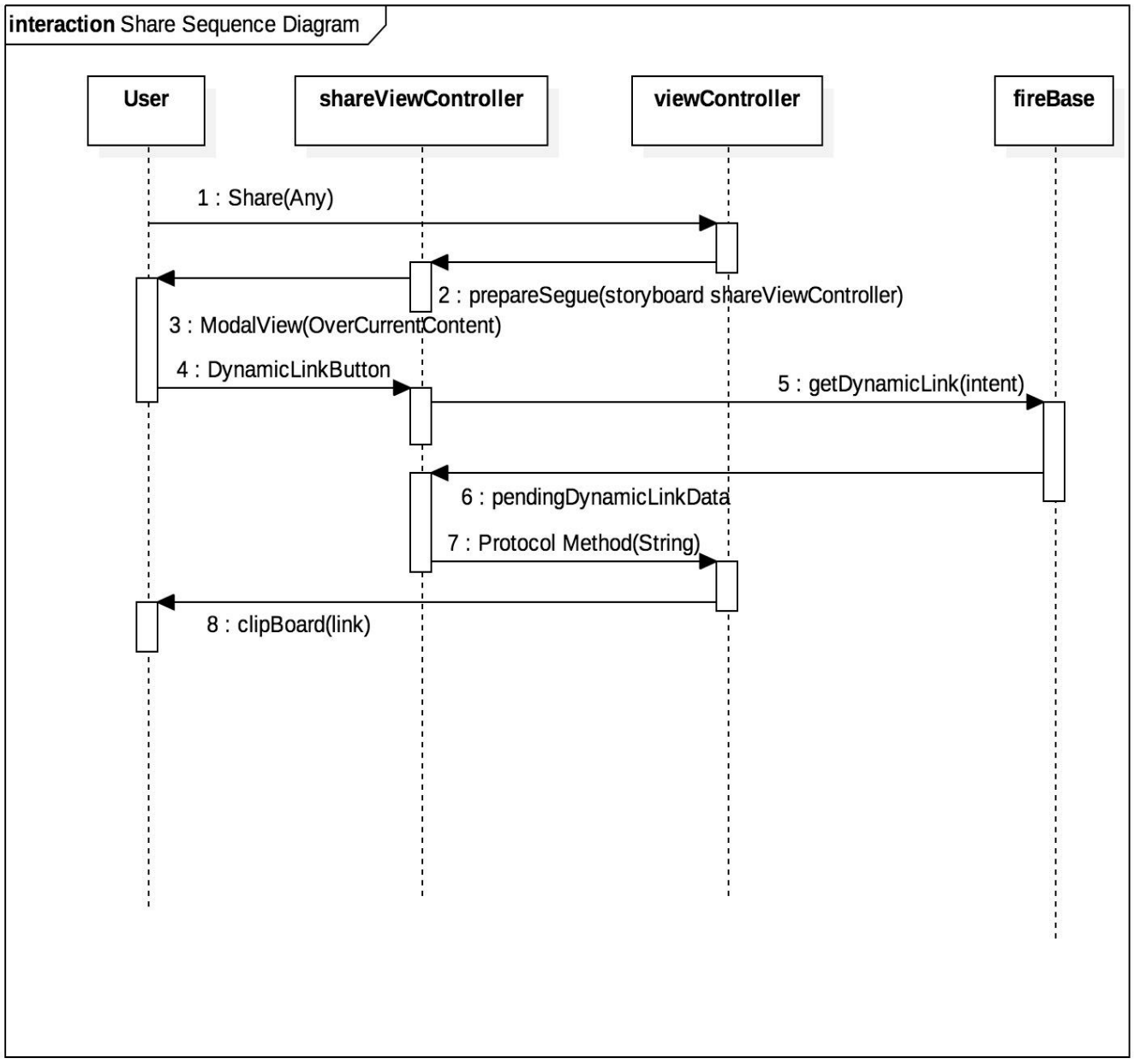


Figure 24 Share SD

4.5 Data Flow Diagram

Level 0

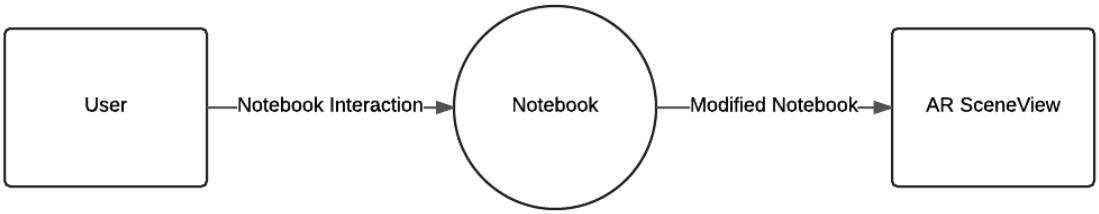


Figure 25 Level 0 DFD

Level 1

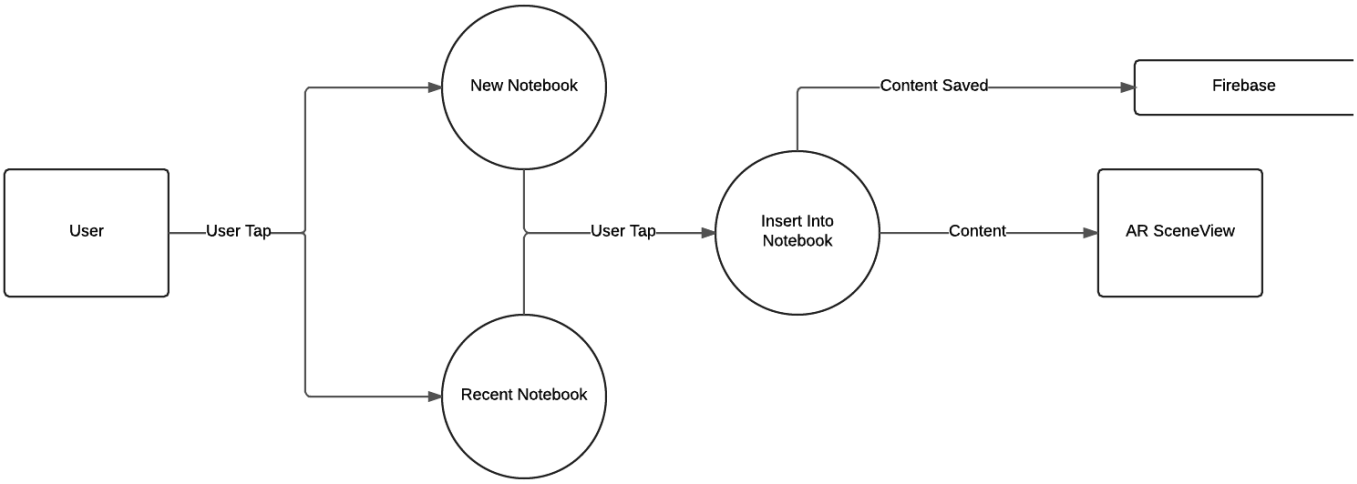


Figure 26 Level 1 DFD

Level 2

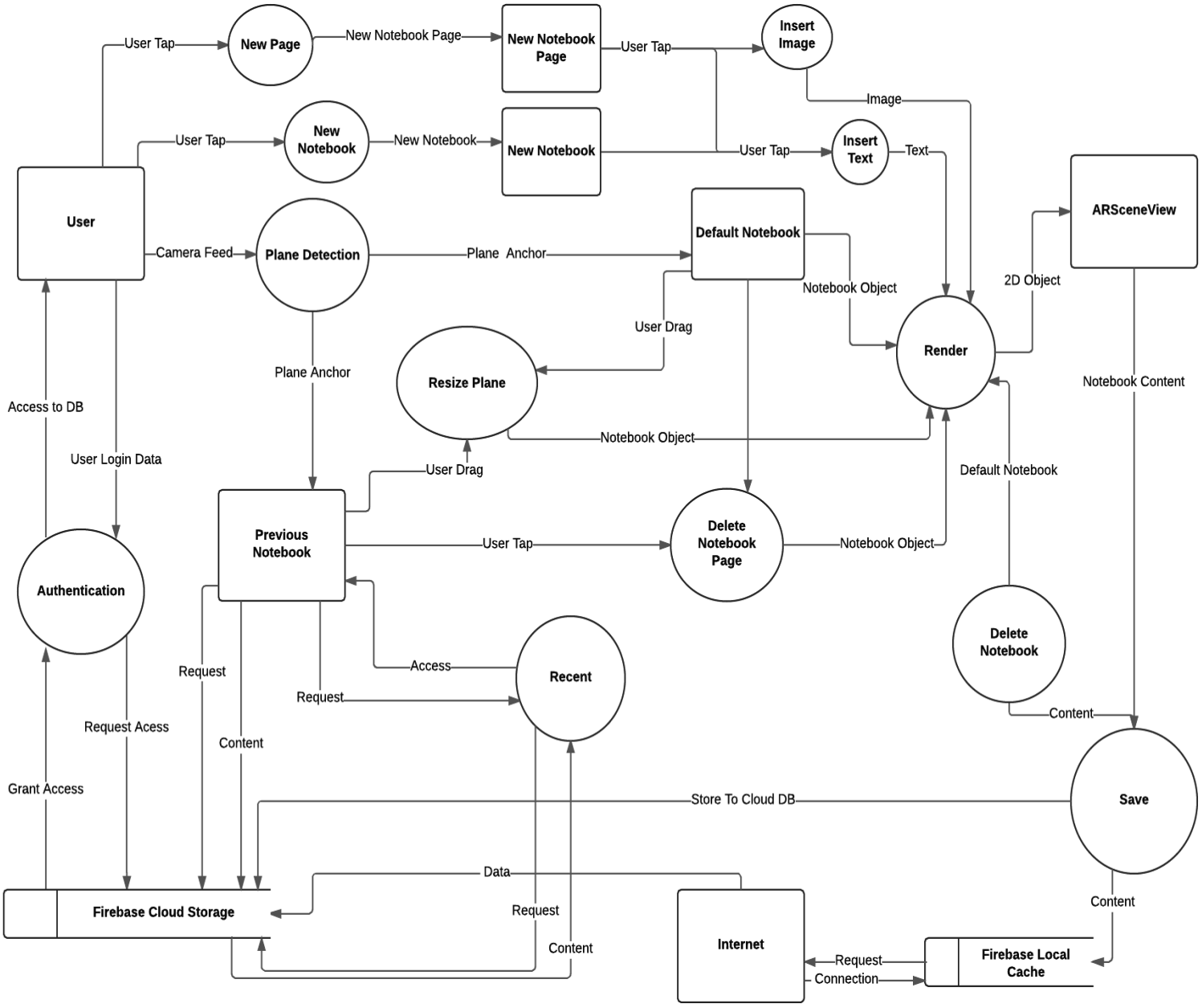
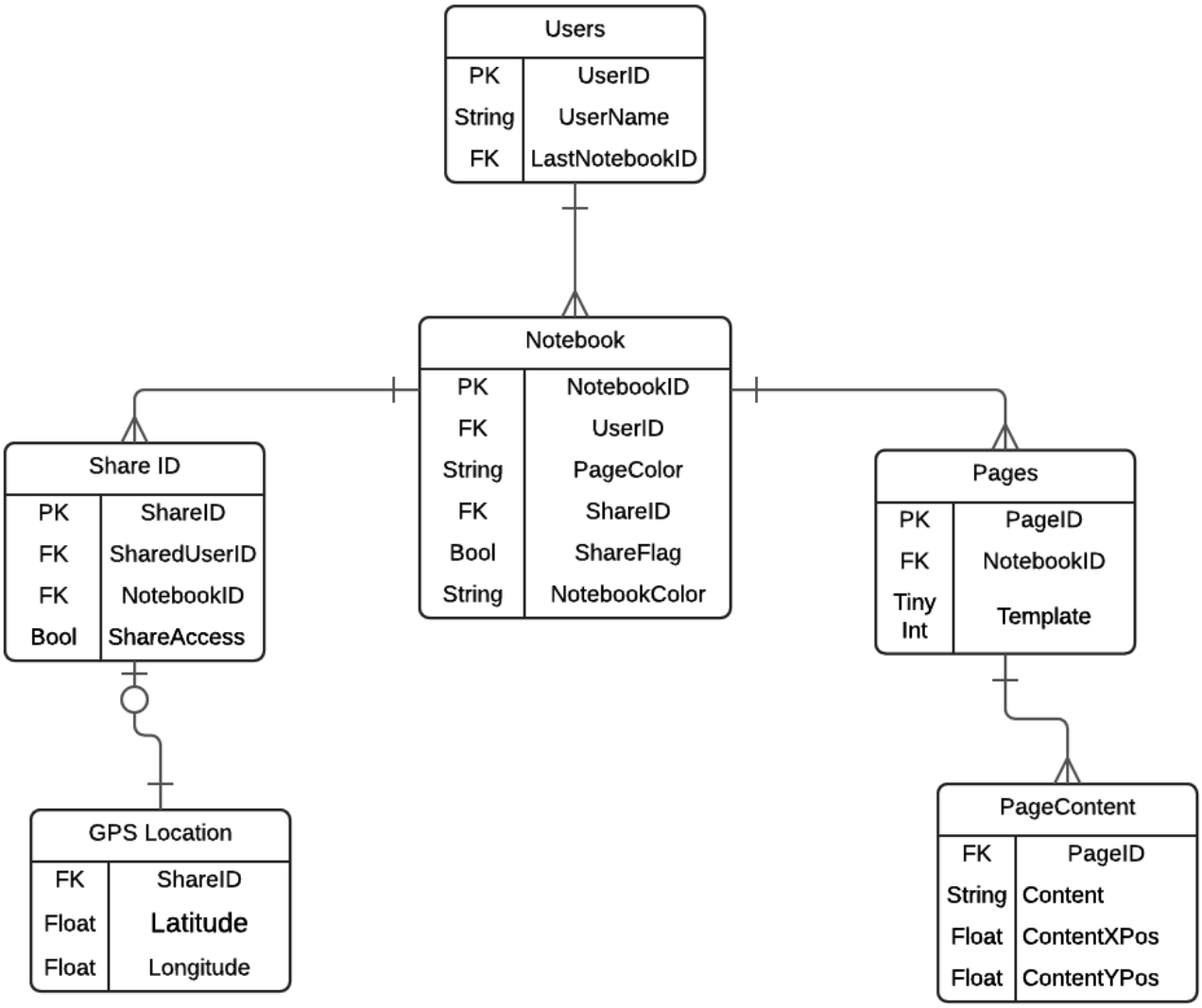


Figure 27 Level 2 DFD

4.6 Database Design



4.7 Class Diagram

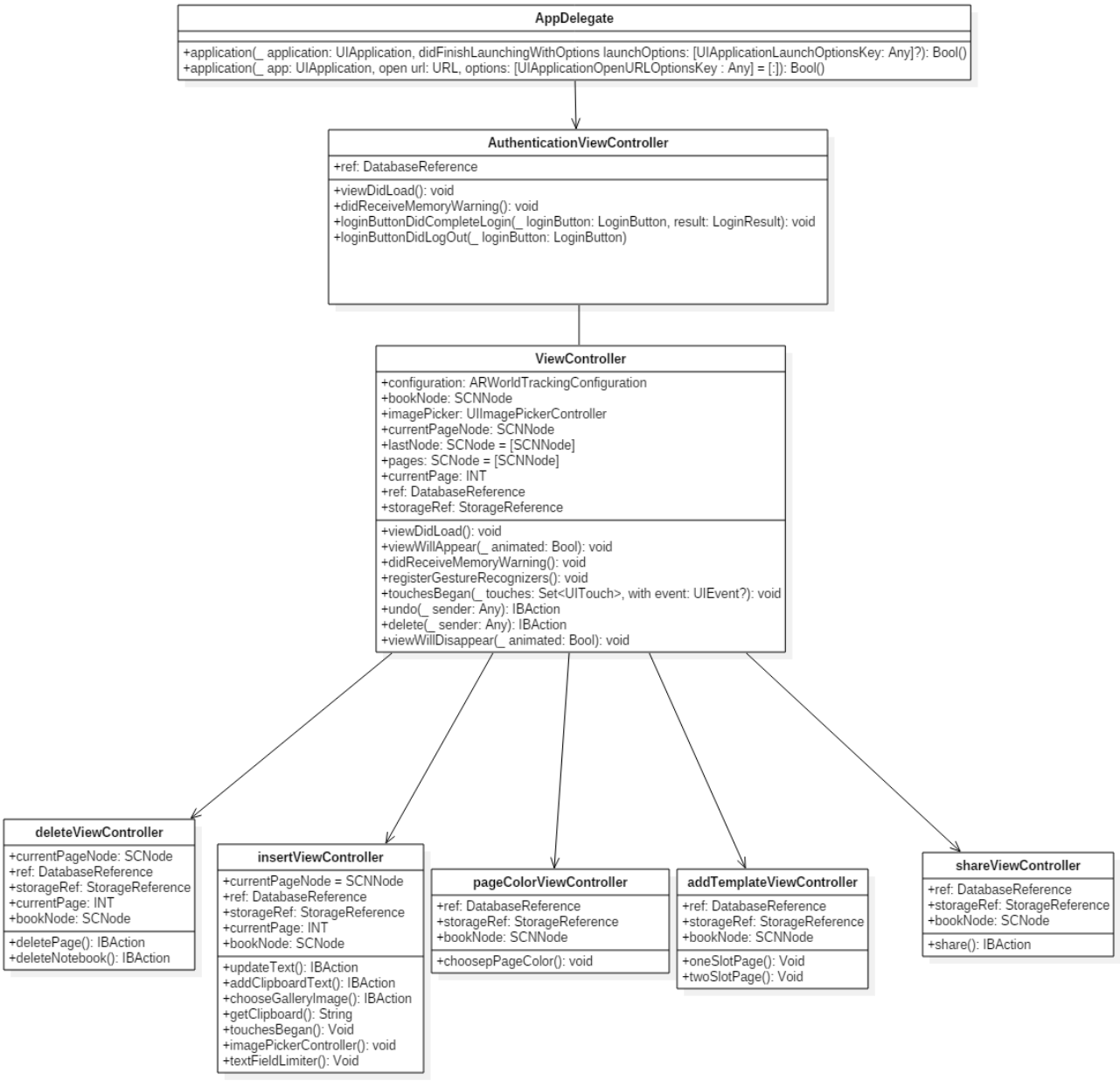


Figure 29 Class Diagram

4.8 Application Program Interfaces

This application uses the following calls provided by Firebase to store and retrieve notebook information. Firebase Authentication Framework is also being used to allow application users to log in using their previously existing Facebook accounts.
Note: The documentation as it currently stands has not yet been updated to support Swift 4. The full documentation can be seen here, and for simplicity we have used the method names listed there. The true syntax in the application may differ.

4.8.1 FirebaseCore

FirebaseCore allows the application to set up a globally accessible database configuration.

Member or Function	Description
FIRApp	This is entry point for all other Firebase methods. It initializes and configures the FirebaseApplicationConnection.

4.8.2 FirebaseAuth

Our application uses methods found in FirebaseAuth. Once the user has authenticated through Facebook they will be granted a authentication token. Some examples of methods or functions our application uses are as follows:

Member or Function	Description
FIRFacebookAuthProvider	Class used to contruct Facebook Credentials
FIRUser	This represents the user currently signed into the application. Used to grab the user's first and last name.

4.8.3 FirebaseDatabase

Our application uses the methods found in FirebaseDatabase to initialize, store and retrieve information from our JSON structured data.

Member or Function	Description
FIRDatabase	The entry point for accessing the application's database. Future references to the database can be used by this reference.
FIRDataSnapshot	After a fetch from the location of FirebaseDatabase information is stored in FIRDataSnapshot to read from.

4.8.4 FirebaseStorage

Our application uses the methods found in FirebaseStorage is used to support the uploading and hosting of images.

Member or Function	Description
FIRStorageMetadata	This class can be used to add information to the image objects before uploading them to storage. We use this to name image files in order to later reference them from the Database.

4.9 User Interface Design

Description: When opening the application for the first time, the user will be prompted to sign in using a Facebook account.

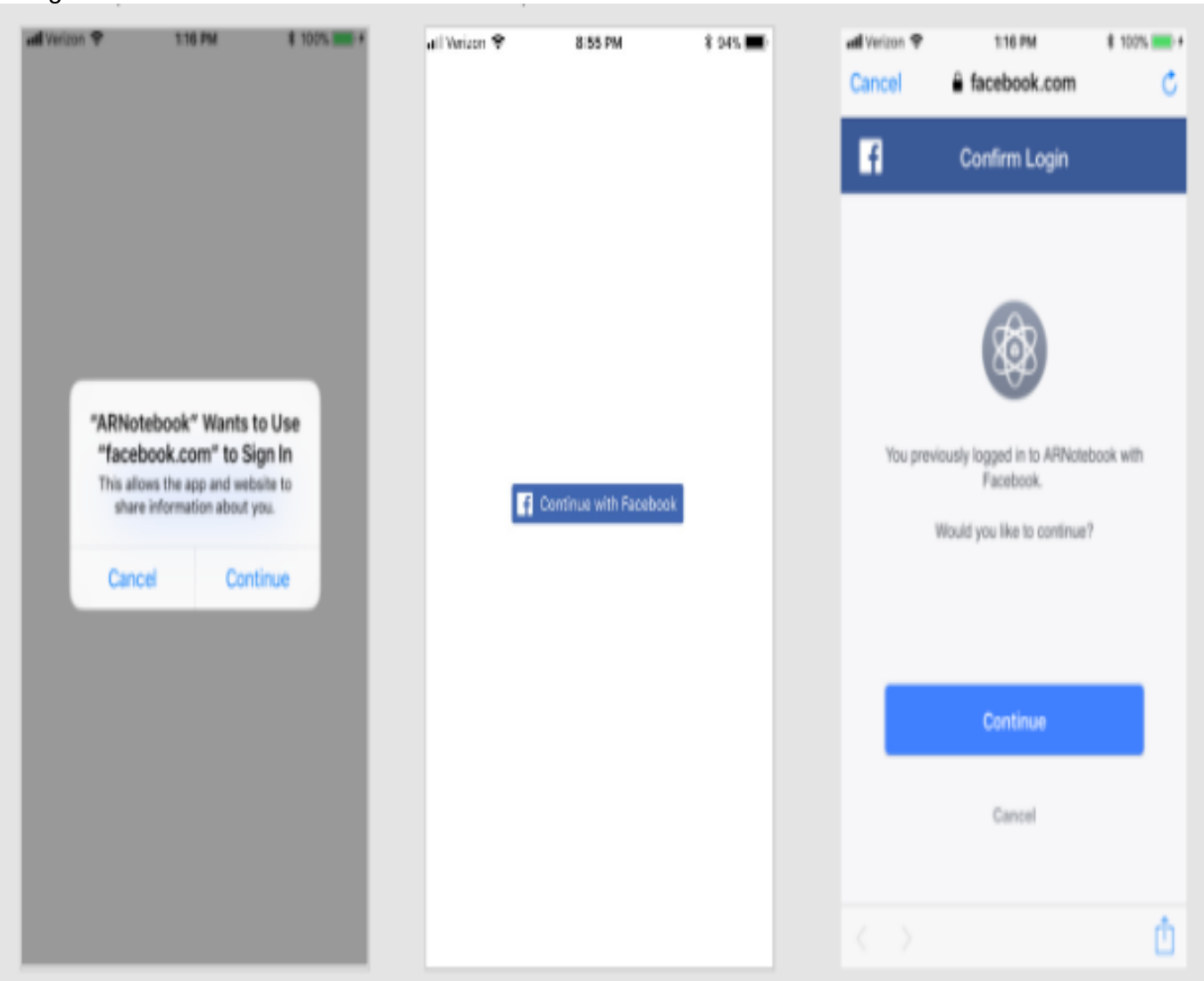


Figure 30 Login UI

Description: After the user has been authenticated, the user must scan the room to obtain a horizontal plane detection. Once the user has successfully obtained the horizontal plane, the user can then tap inside the plane to place a default notebook.

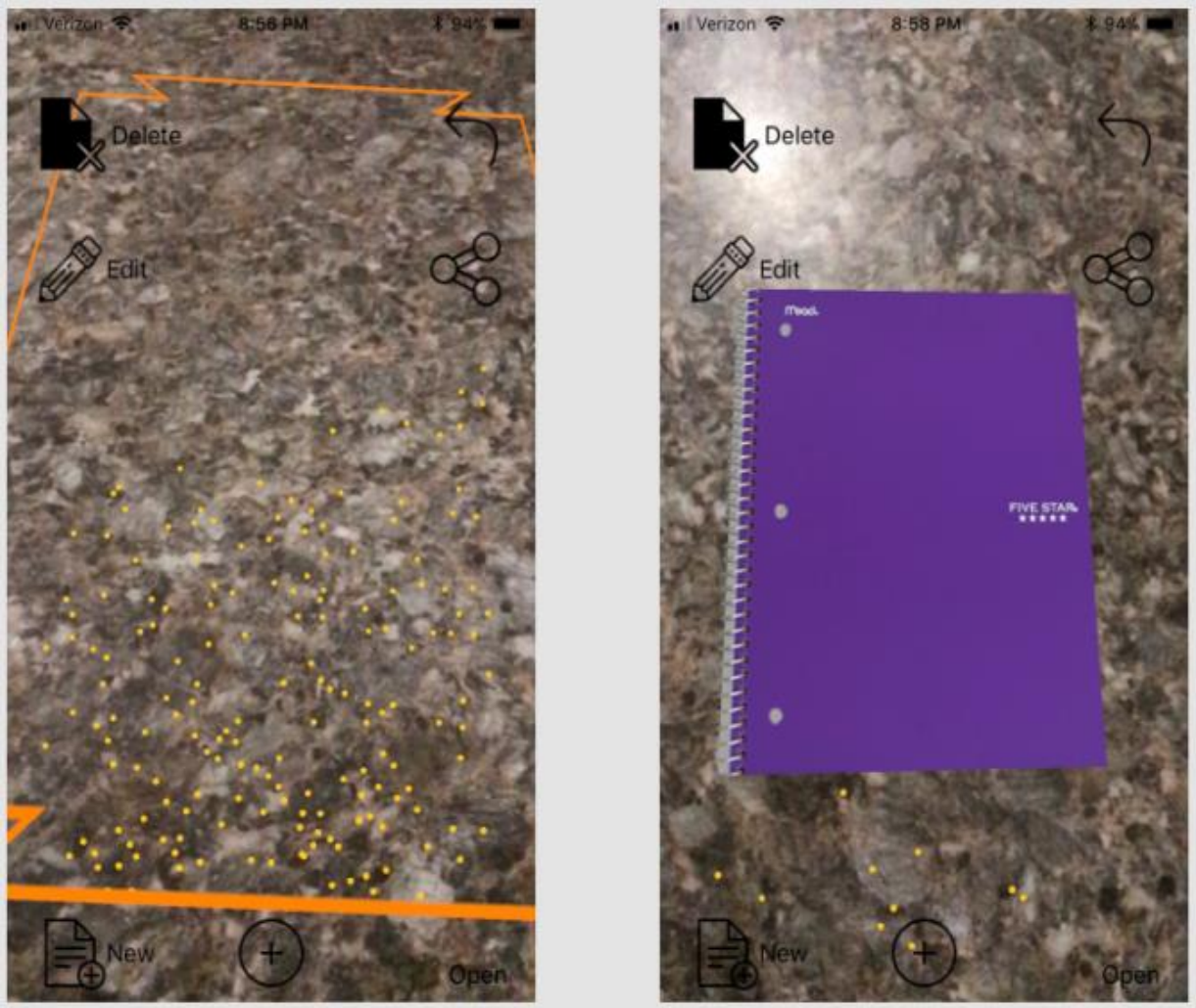


Figure 31 Plane Detection UI

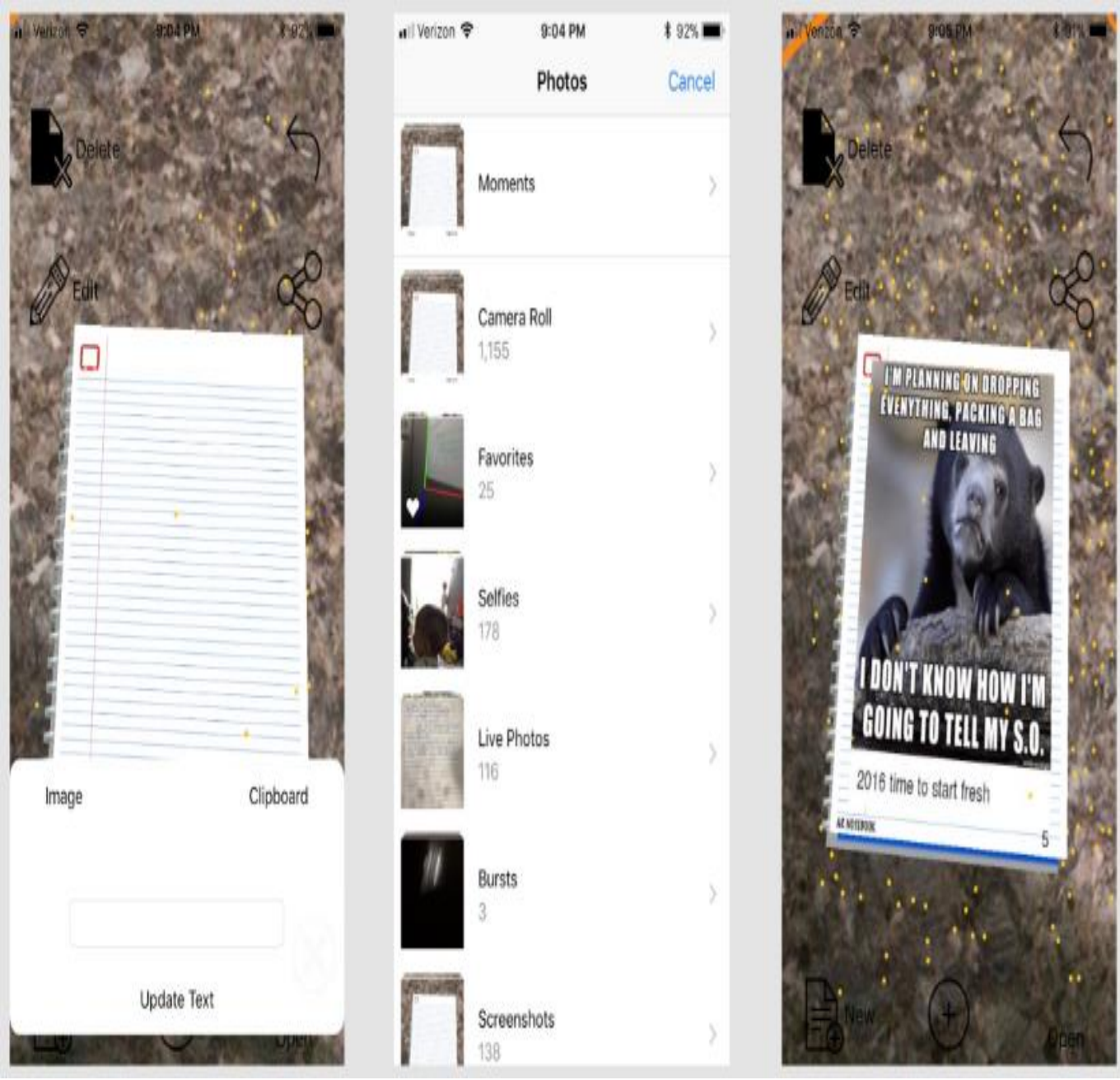
Description: After the user has added a notebook, the user then can add pages. Once the user selects the new page button, a menu for templates displays. The user must choose one or two page templates.

Figure 32 Add Page UI

Description: After the user has added a notebook a page, the user has the option to change the page color.

Figure 33 Change Page Color UI

Description: One the user has a page, the User can add an Image, text, or text saved from the user's clipboard.



Description: Once the user has an image, or text on the page, the user can then access the undo button.

Figure 34 Undo UI

Description: The user has the option to delete the notebook or page. once the user taps the delete option, the pages or the notebook will be deleted.

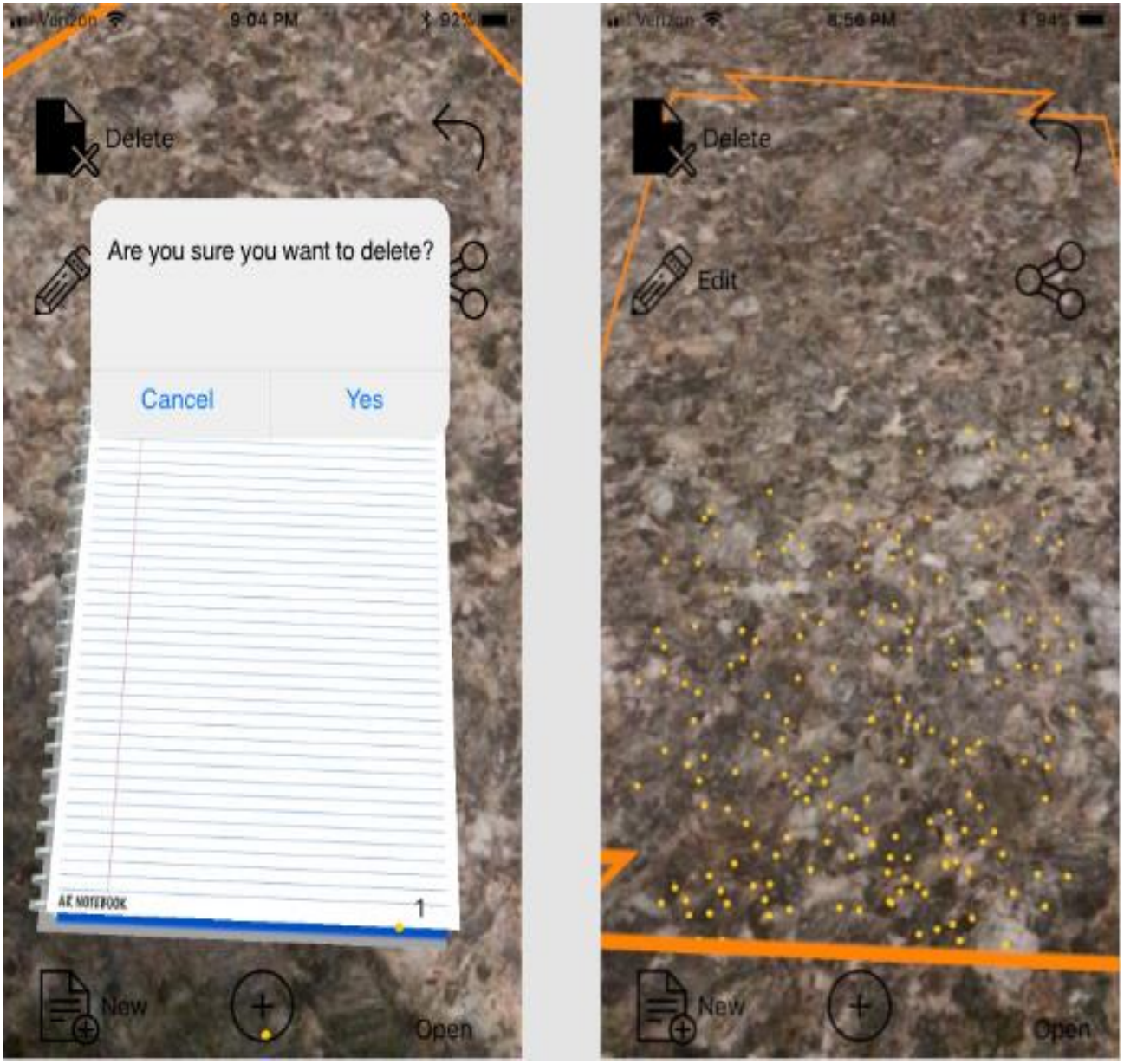


Figure 35 Delete UI

Description: The Open Option allows the user to open a new notebook or retrieve a previous notebook.

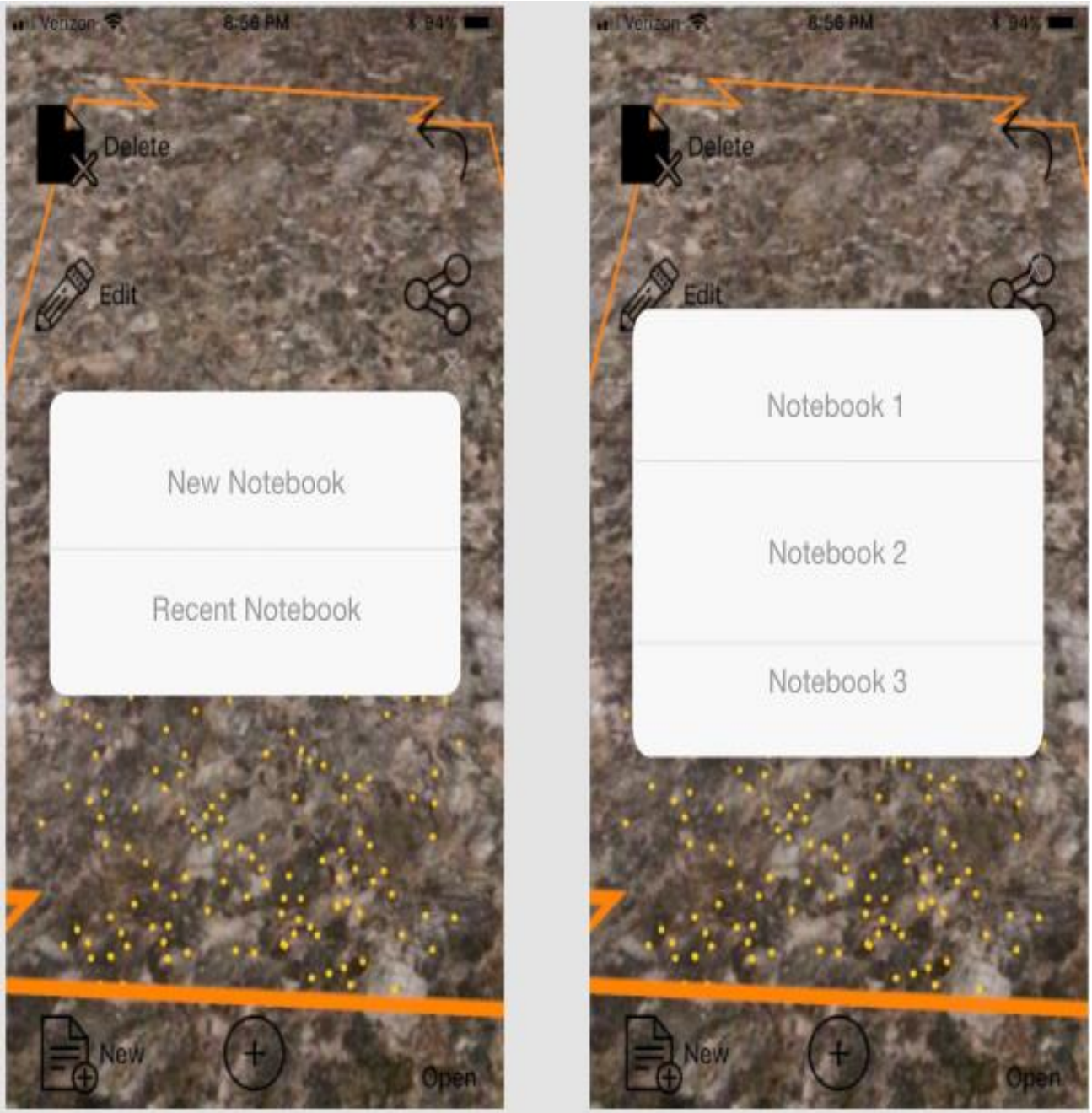


Figure 36 Open Notebooks UI

Description: The Share Option allows the user to share a notebook with another user

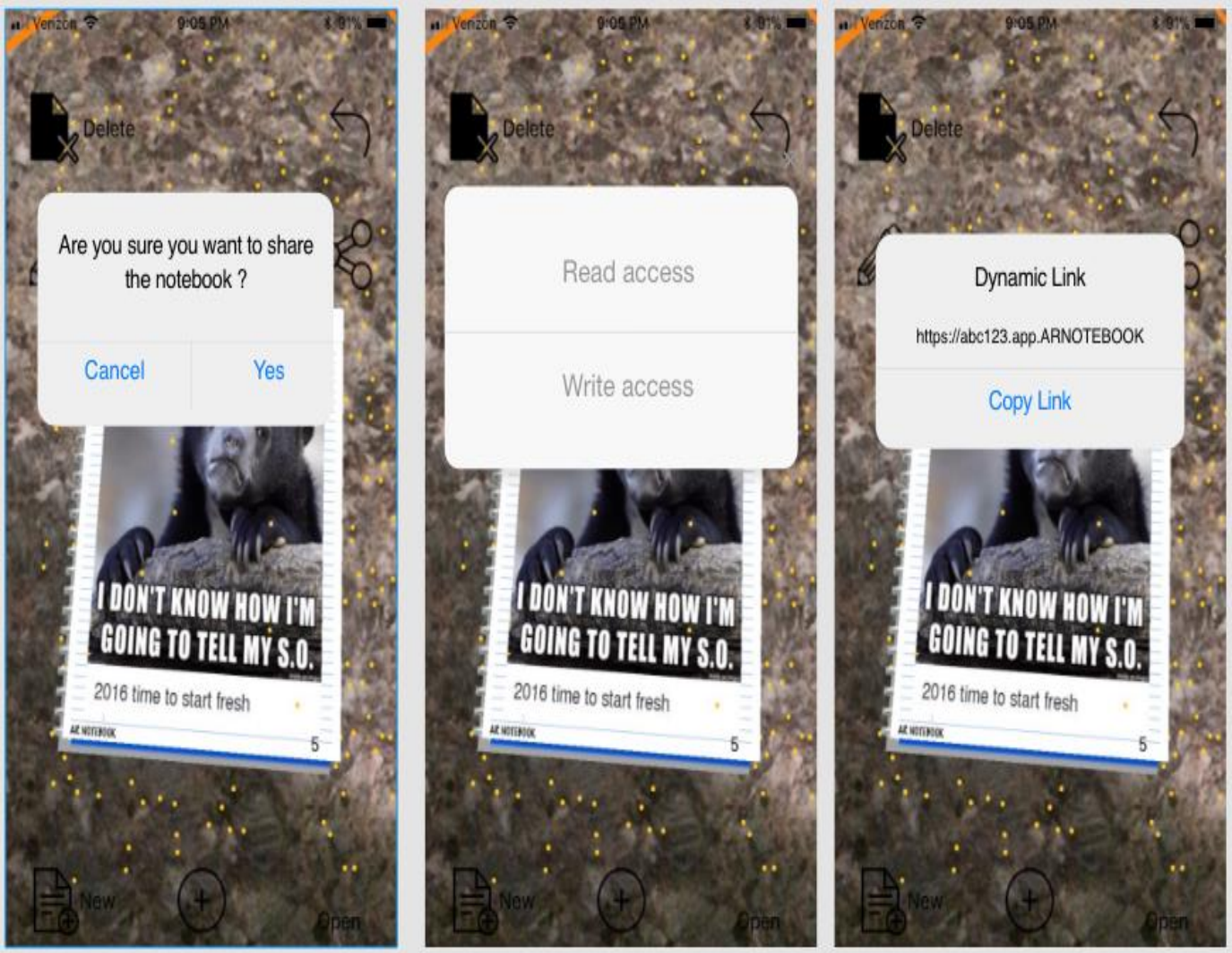


Figure 37 Share UI

5 Product Design Specification Approval

The undersigned acknowledge they have reviewed the AR Notebook Team **Product Design Specification** document and agree with the approach it presents. Any changes to this Requirements Definition will be coordinated with and approved by the undersigned or their designated representatives.

Signature: _____ Date: _____

Print Name: _____

Title: _____

Role: _____

Signature: _____ Date: _____

Print Name: _____

Title: _____

Role: _____

Signature: _____ Date: _____

Print Name: _____

Title: _____

Role: _____

Appendix

Requirements Traceability Matrix

Requirement ID	Requirement Name	Use Case ID	Test Plan ID
FR3.2.1	**User Authentication**	UC-1	
FR3.2.2	**Welcome and Instruction Message**	All Use Cases	
FR3.2.3	**Create a New Notebook**	UC-2	
FR3.2.4	**Create a New Notebook Page**	UC-4	
FR3.2.5	**Insert Text**	UC-6	
FR3.2.6	**Insert Text from the Clipboard**	UC-5	
FR3.2.7	**Insert Image from Gallery**	UC-7	
FR3.2.8	**Turn Notebook Page**	UC-3 through UC-10	
FR3.2.9	**Save the Notebook**	UC-2 through UC-10	
FR3.2.10	**Delete a Notebook Page**	UC-9	
FR3.2.11	**Delete a Notebook**	UC-8	
FR3.2.12	**Undo**	UC-10	
FR3.2.13	**Resize Image**	UC-7	
FR3.2.14	**Retrieve the Notebook**	UC-3	

FR3.2.15	**Choose from Page Template**	UC-4	
FR3.2.16	**Sharing Notebooks**	UC-11, UC-12	
FR3.2.17	**GPS Notebook Locations**	UC-13	
FR3.2.18	**Alert the User on Low Battery**	UC-14	