Software Requirements Specification

AR Notebook

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

Darion Thompson

Revision History

Date	Revision Number	Description
9/26/17	Version 1.0	First Draft
9/28/17	Version 1.1	Update Requirements with Wireframes
9/30/17	Version 1.2	Updated Requirements, Overviews, Rough Draft of DFD
10/4/17	Version 1.2.1	Updated wireframes with new single-page wires
10/5/17	Version 1.3	Updated DFD, Removed functional requirements specific to UI Functions
10/7/17	Version 1.4	Updated Requirements and overviews
10/8/17	Version 2	Finalized Document

Document Approval

Signature	Printed Name	Title	Date
Megan Majewski	Megan Majewski	Team Lead	10-8-17
Darion Thompson	Darion Thompson	Front-end Lead	10-8-17
Mohammed Ali	Mohammed Ali	QA Lead	10-8-17
Artur Bushi	Artur Bushi	Back-end Lead	10-8-17
	Erfan Najmi	Client	

Table of Contents

Table of Figures

1.0 Introduction

1.1 Purpose

The purpose of this software requirements specification document is to provide a detailed description of the required functionality of the Augmented Reality (AR) Notebook as discussed between the development team and the client. This document will also give an overview of the applications features as well as the hardware, and software dependencies.

1.2 Scope

The AR notebook application is an augmented reality iOS application built using Apple's ARKit. This project is designed to make interacting and viewing notebooks more engaging.
Upon completion of the project, this application will be available to be download in Apple's App Store and useable on any iPad or iPhone running on iOS 11.0 or higher with at least an A9 processor unit.
The AR notebook application will allow registered users to view, create, save, and interact with a virtual notebook. These interactions will include opening notebooks, inserting images, inserting clipboard items, saving, and retrieving notebook information.

1.3 Definitions, Acronyms, and Abbreviations

Term	**Definition**
Augmented Reality (AR)	A technology that superimposes a computer-generated image on a user's view of the real world, thus providing a composite view.
iOS Application	A mobile application that is specific to Mac products. (iPhone, iPad)
SceneKit	An application framework that is responsible for rendering three dimensional objects.
ARKit	A framework that uses the iOS device's camera and motion features to create the augmented reality scene.
UIKit	A framework that is responsible for creating and managing the user interface.
App	Application

Photos	iOS term that refers a device's photo gallery
Database	A collection of information that is organized in a format that can be accessed. The information can be managed by adding, deleting, and updating.
Firebase	The database that will be used to save the applications information
pod ‘Firebase/Core’	Prerequisite libraries and Analytics
JSON	JavaScript Object Notation. It is used to transmit data between server and application.
Scene Text Node	3D model that is made up of text
App Store	A digital store where users can download and install desired mobile applications or content to their devices
SceneView	The view in the augmented reality application
Clipboard	Temporary phone storage used for copying and pasting small amounts of data
Apple	American technology and innovation company
DB	Shorthand for database

1.4 References

[1] *Apple Developer Documentation*, Apple, developer.apple.com/documentation. Accessed 30 Sept. 2017.
[2] Percival, Rob, and Nick Walter. “The Complete iOS 11 & Swift Developer Course - Build 20 Apps.” *Udemy*, 24 Aug. 2017, www.udemy.com/complete-ios-11-developer-course/learn/v4/content. Accessed 30 Sept. 2017.
[3] Slim, Rayan. “IOS Augmented Reality - The Complete Course on ARKit.” *Udemy*, 20 Sept. 2017, www.udemy.com/ios-augmented-reality-the-complete-course-on-arkit/learn/v4/overview. Accessed 30 Sept. 2017.

1.5 Overview

The remainder of this document will cover three major sections. The first section is the general design overview of the project and several prerequisites. The second section covers the functional and nonfunctional requirements needed to produce this app in full. The requirements will state the conditions and assumptions that are needed to complete the requirement. The third section will display data flow diagrams to go in depth on how the data flows whenever the user interacts with the application. Following the three sections will be a section regarding wireframes and the User Interface as well as an Appendix containing the requirements traceability matrix.

2.0 General Description

This section will provide an overview of the AR Notebook application system. Each section will contribute to the overall functionality of the application. This General Description section goes in depth about how the app is one of the first of its kind in development, and has been developed from scratch. Finally, there will be an overview that will describe the functions of the app, user characteristics, general constraints, and assumption and dependencies.

2.1 Product Perspective

The AR notebook application is an innovation based on apple's ARKit. ARKit combines two separate components into one application. The first component is the application's functionality, which will be like that of Google's Keep, or Evernote application. For example, the user will be able to create notebooks to keep track of notes as they desire. The second component of this application consist of integrating Augmented Reality technology to the existing application notebook functions. The application will use the built-in iPhone or iPad camera to gather a real-time view of the world. Next, the notebooks for the user to interact with will be displayed. In this area, the AR Notebook is an innovation, and does not have a similar augmented reality product in the market to compare to.

2.2 Product Functions

This application gives registered users the ability to create, read, edit, and save notebooks. The main functionality of this application is based on the ability to edit objects in the AR world. Notebooks will be placed on horizontal surfaces and can be viewed through the device's camera and interacted with within the application. This interaction will come in a variety of features. The user input will consist of accessing the clipboard content and images to edit and add content to the notebook. The users will also be able to interact with the items placed on top of the notebook pages or templates by dragging to resize images or text. The pages of the notebook will be turned by the user swiping the screen left or right.

2.3 User Characteristics

This mobile application is intended for personal use on iOS devices. The application will be available to customers through the App Store so long as their devices are running iOS 11 or higher, have a A9 processor chip and higher. These software and hardware requirements are essential for the support of ARKit.

2.4 General Constraints

To use this application, the user must be authenticated through Facebook login, which makes our application reliant on a stable internet connection.

2.5 Assumptions and Dependencies

This section will cover the assumptions and dependencies that are expected and required of the user.

2.5.1 Assumptions - User iOS Application

It is assumed that the user's iOS application will be used on an iOS version 11 or higher. The devices are expected to be an iPhone or an iPad that has an A9 processor chip or above. The devices are expected to have a stable internet connection to authenticate into the application, and share content from there. The physical mobile devices are expected to have a functional touch screen which is essential for processing and handling the user's interaction. The device must also have a functional camera to be able to scan the user's environment and connect reality the augmented reality. It will also be assumed that the user knows how to utilize the device. It will also be assumed that the user knows how to access the App store as well as how to download the app.

3.0 Specific Requirements

This section contains information regarding the external and internal interface requirements and hardware interfaces.

3.1 Interface Requirements

3.1.1 User Interfaces

AR Notebook is an iOS mobile application which will be available for all devices running iOS11 and higher with at least an A9 processor. Apart from the user authentication, this application is designed to be a single screen with multiple functions. After logging in with Facebook the user will be presented with a notebook. This notebook will be a default blank notebook, or if the user has a notebook they created before it will be the last notebook that they had opened.

The notebook will be customizable: users will be able to choose the style of the notebook cover and a style of a template for their notebook pages. The templates will have pre-defined areas where the user can place their content. The content the notebook supports is images imported from Photos, or text taken from the keyboard or clipboard. The text or image items placed into these template pages should be adjustable through touch screen dragging, giving the user the ability to change the size of the images or text. As the user adds their desired content to the screen it is auto-saved to the database for future opens of this notebook.

The background color of the notebook pages will also be customizable. At any point, the user will be able to see a 360°, left-to-right view of their notebook object placed in AR.

Below is a visual illustration of user interface and the many interactions available:

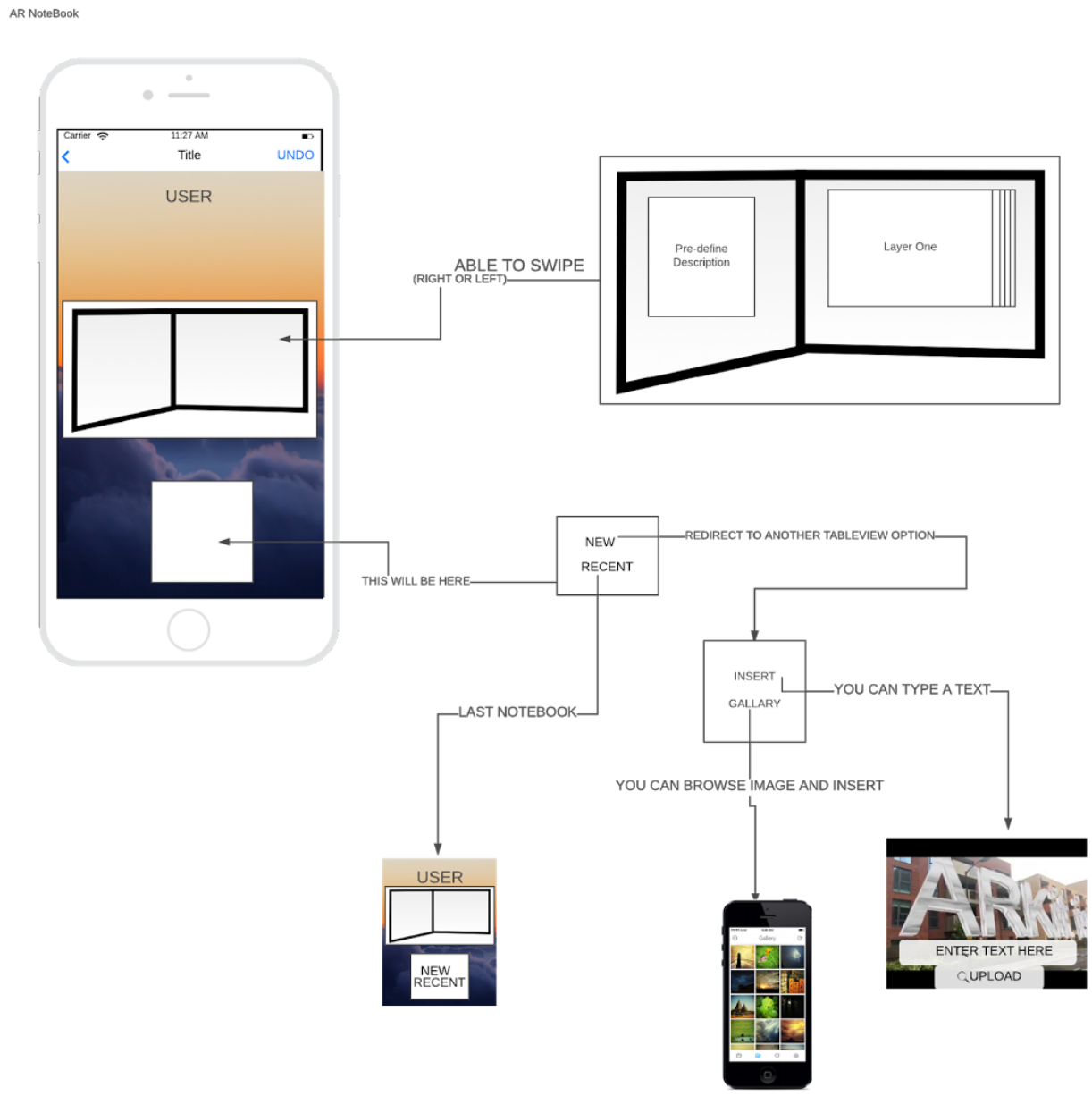

Figure 1: Visual Interface

3.1.2 System requirement:

- iPhone 6s and 6s Plus
- iPhone 7 and 7 Plus
- iPhone SE
- iPad Pro (9.7, 10.5 or 12.9) – both first-gen and 2nd-gen
- iPad (2017)
- iPhone 8 and 8 Plus
- iPhone X

3.1.3 Software Interfaces

The AR Notebook application will require ARKit and SceneKit.

3.2 Functional Requirements

The system requirements apply to each of the following functional requirements.

FR3.2.1 User Authentication

Description: The application will require the user to log into an account. The account must be a verified Facebook account to see the application features and data. The user will have to login to the application each time they open the app.
Processing: Firebase Facebook Authentication
Precondition: The user has clicked the AR Notebook app to start the application and they have a Facebook account.
Postcondition: The user is authenticated and admitted into the application.

FR3.2.2 Welcome and Instruction Message

Description: After logging into the application through Facebook a notification will be displayed to welcome the user. The application will then show a message that will instruct the user on how to scan the room. It will also inform the user on what they will see (feature points), and if the camera can detect any surfaces.
Processing: Displaying a message popup.
Precondition: The user has been authenticated.
Postcondition: The camera detects the surface needed to place the notebook.
Assumptions: The camera is fully functional.

FR3.2.3 Create a New Notebook

Description: The user can create a new notebook object. The notebooks will have three predefined cover styles. Each cover style will vary in colors, images and text.
Processing: Processing happens at the next forward dependency.
Precondition: The notebook.scn models are available.
Postcondition: The user has the option to choose a notebook cover style.

Forward Dependency: FR3.2.4

FR3.2.4 Create A New Notebook Page

Description: At any time, the user can choose to create a new blank, or templated notebook page by choosing the option on the menu.
Processing: N/A
Precondition: The notebook must be created and open.
Postcondition: A new page with the template of the notebook is added into the notebook.
Forward Dependency: FR3.2.2

FR3.2.5 Insert Text

Description: When the user clicks the add from keyboard option, the default iOS keyboard will show on the screen. This will allow the user to enter in their own set of characters to the notebook object. Clicking off the keyboard onto the screen, or pressing the return button will close out of the keyboard view and render the input text onto the notebook page.
Processing: Text Field responder to capture user's input text and pass through to SCNText creator.
Precondition: N/A
Postcondition: The notebook object is updated with the content which is placed on the page.

FR3.2.6 Insert Text From the Clipboard

Description: When the user taps the clipboard menu option the latest content inside the user's pasteboard is rendered on the current page based on the chosen slot in the notebook page template. If no notebook page template has been previously chosen by the user the pasteboard text will be inserted at the center of the page.
Processing: The application will access the User Interface Pasteboard function. This function allows access to the user's clipboard. It sends the content to the Scene Text Node.
Precondition: The content is in string format only.
Postcondition: The notebook object is updated with the content which is placed on the page.

FR3.2.7 Insert an Image From Gallery

Description: When clicking the gallery menu option this function will allow the user to choose to import an image from their gallery photos. The selection will open the default iOS Photos viewer where the user can select an image to import to the notebook page. The image will be rendered on the current page based on the chosen slot in the notebook page template. If no notebook template is chosen the image will be placed at the center of the page.
Processing: The application will access the User Interface Image Picker function. This function is responsible for accessing the photo library and returning the image from UIImagePickerController.
Precondition: The user's device has at least one photo saved in its photos gallery. The image file must be less than 10mb.

Postcondition: The selected photo will be rendered on the notebook page.

FR3.2.8 Turn Notebook Pages

Description: The default view loaded of a notebook is the notebook cover. The user will be able to interact with the notebook to turn the pages by swiping on top of the notebook page object left or right. After the cover page, the notebook will show the next page in the notebook. Only one single page will be shown at a time.
Processing: The application will utilize the Tap Gesture Recognizer function to sense when the user is touching the notebook.
Precondition: The notebook object has pages inside of it.
Postcondition: The next page of the notebook is shown.

FR3.2.9 Save the Notebook

Description: This application features auto-save functionality. Each time that an item is inserted into the page, the content will be saved to the database user that user's profile automatically.
Processing: Firebase database will store the string and chosen images in a data table unique to that notebook.
Precondition: There is an open, authenticated connection to the database.
Postcondition: The content of the notebook is successfully stored.

FR3.2.10 Delete a Notebook Page

Description: When the user taps the delete option, an alert to confirm deletion will appear. If the delete action is confirmed, the page that was added to the screen will be removed.
Processing: The application will drop the notebook page and its content from the Scene View and the database.
Precondition: The notebook contains some information on the page that can be removed.
Postcondition: The notebook is rendered without the last note visible.

FR3.2.11 Delete a Notebook

Description: The user can delete the entire existing notebook. When the user taps the delete option, an alert to confirm deletion will appear. If the delete action is confirmed, the page that was added to the screen will be removed.
Processing: The application will drop the notebook and its content from both the SceneView and the database.
Precondition: The notebook information exists in the database.
Postcondition: The notebook information is dropped from the database

FR3.2.12 Undo

Description: The user can undo the single last insert action they performed, whether it be an insert image, insert text, or insert from clipboard. Delete actions after confirmation cannot be undone.
Processing: Removing the last child node. Updating database to drop node.
Precondition: There was a node added to the notebook page.
Postcondition: The last item inserted is not shown on the notebook.

FR3.2.13 Resize Image

Description: The user has the ability to resize the image dimensions by dragging the edge. The image must stay contained within the open notebook page, but can stretch to be the entire width of the page.
Processing: Recognizing pinch and drag gestures from user, function created by authors that resize the coordinates based on user's finger location.
Precondition: Image already place in the view by user.
Postcondition: The content will be displayed on the screen at the newly chosen size.

FR3.2.14 Retrieve the Notebook

Description: If the user wants to open a previously saved notebook, they have the option to select from a list of previously all saved notebook objects.
Processing: Fetch the content information from the database. The content will be loaded back to the notebook in the same order that it was placed. The items inside the pages will maintain their original placement from when the notebook was created.
Precondition: The user has already created and saved at least one notebook.
Postcondition: The content is pulled into as a collection.

FR3.2.15 Choosing from Page Templates

Description: When choosing to add a page the user can also elect to choose a template for that page. The page templates will have predefined slots inside of them in which text or images can be placed. This allows the user the ability to add more than one type of content to a page. The application will have two different page templates to choose from. One page template has one container to which you can add any one item. The second template option will have two different containers to choose from.
Processing: A new page of that template is added to the notebook object.
Precondition: A notebook is opened in the application
Postcondition: The newly chosen blank template is rendered.

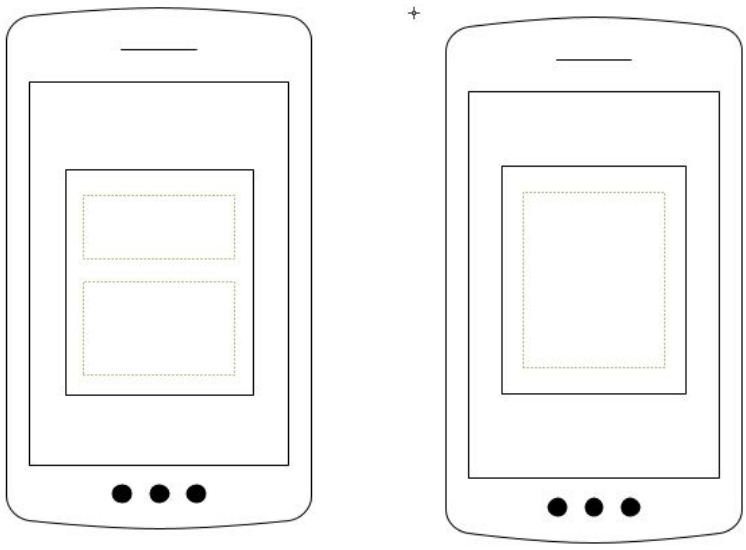

Figure showing 2 Template Options

FR3.2.16 Sharing Notebooks

Description: Users with saved notebooks will have the ability to share their content with other users.
Processing: Firebase dynamic link sharing.
Precondition: Database already connected with notebooks saved inside.
Postcondition: The notebook sent to a different user will be read only access to that notebook.

FR3.2.17 Stretch Goal: GPS Notebook Locations

Description: Users will have the option to drop a pin with a specific notebook at a location they choose. This GPS pin will allow other users of the application to view the notebook placed at the application and make edits to ti.
Processing: GPS Location for notebook and the user when using the application.
Precondition: The user has their location services turned on and the application can access their geolocation coordinates.
Postcondition: The chosen notebook will be give GPS coordinates and can be viewed by anyone who is near the area running the application.

FR3.2.18 Alert the User on Low Battery

Description: ARKit requires the user's device to have at least 5% battery, When the application detects that the batter is at or below 5%, the user will get an alert that notifies them of their low battery alert. The user will be able to exit out of this alert if and only if they are above 5% battery. Otherwise, the alert will keep reappearing.

Processing: Configure the battery monitor to check for battery power on start of the application.
Precondition: N/A
Postcondition: The alert is displayed, and the application is shut down.

3.3 Non-Functional Requirements

NFR3.3.1 Notebook Security

Description: Each user of the application must be authenticated through Facebook. Each user's notes can be viewed only by that user when they are authenticated into the application, unless otherwise shared. In the event the notebook is shared only the single shared notebook can be seen by an alternate user.

NFR3.3.2 Notebook Maintainability

Title: Notebook Maintainability
Description: With future updates of ARKit, this application will be able to integrate new features as they are released.

NFR3.3.3 Notebook Portability

Description: This application will be accessible for anyone with access to the App Store.

3.4 Design Constraints

3.4.1 Database Constraints

Firebase database will be used for this project. Firebase is a cloud based database that is sorted in JSON. User authentication happens through Firebase, and handles the storage of data. Once the user is authenticated, the database synchronizes data in real-time so long as the client has a stable internet connection. This database will be essential for not just accessing and sharing traditional data, but data that contains image data or files.

DBC1 Image Size Limitation

Title: Image Size Limitation
Constraint: Firebase image cannot be over 10MB.

DBC2 Connection Request Limitation

Title: Connection Request Limitation
Constraint: Our Firebase subscription allows 100 Simultaneous connections to the database.

DBC3 Update Capabilities Limitation

Title: Update Capabilities Limitation
Constraint: Our Firebase subscription allows 90,000 create, read, update, or delete operations per day.

DBC4 Data Storage Limitation

Title: Data Storage Limitation
Constraint: Our Firebase subscription allows 5 GB of total stored data.

3.4.2 Application Constraints

ADC1 Internet Connection Limitation

Title: Internet Connection Limitation
Constraint: When the user has no internet connection, the max capacity for data storage is the size of the local cache.

ADC3 Lighting Limitation

Constraint: The User needs adequate light to use this application. A surround light should be at least 4,000k white balance for the camera to properly read the user's surroundings.

ADC4 Flat Surface Constraint

Constraint: To place an augmented notebook, the user must have a flat surface that can be visible through the device's camera.

4.0 Analysis Models

Level 0 DFD

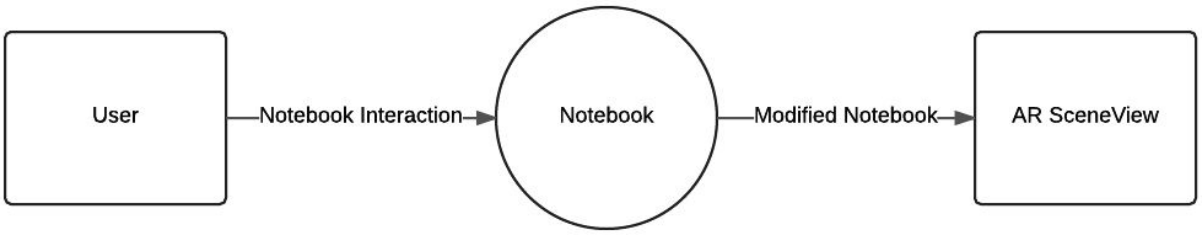

Figure 2: Level 0 DFD

Level 1 DFD

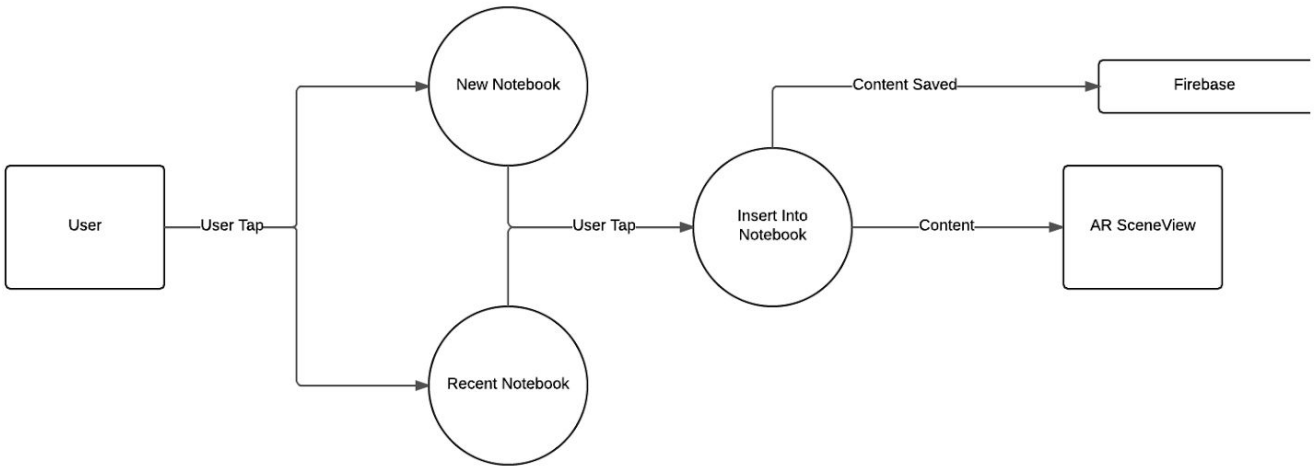

Figure 3: Level 1 DFD

Level 2 DFD

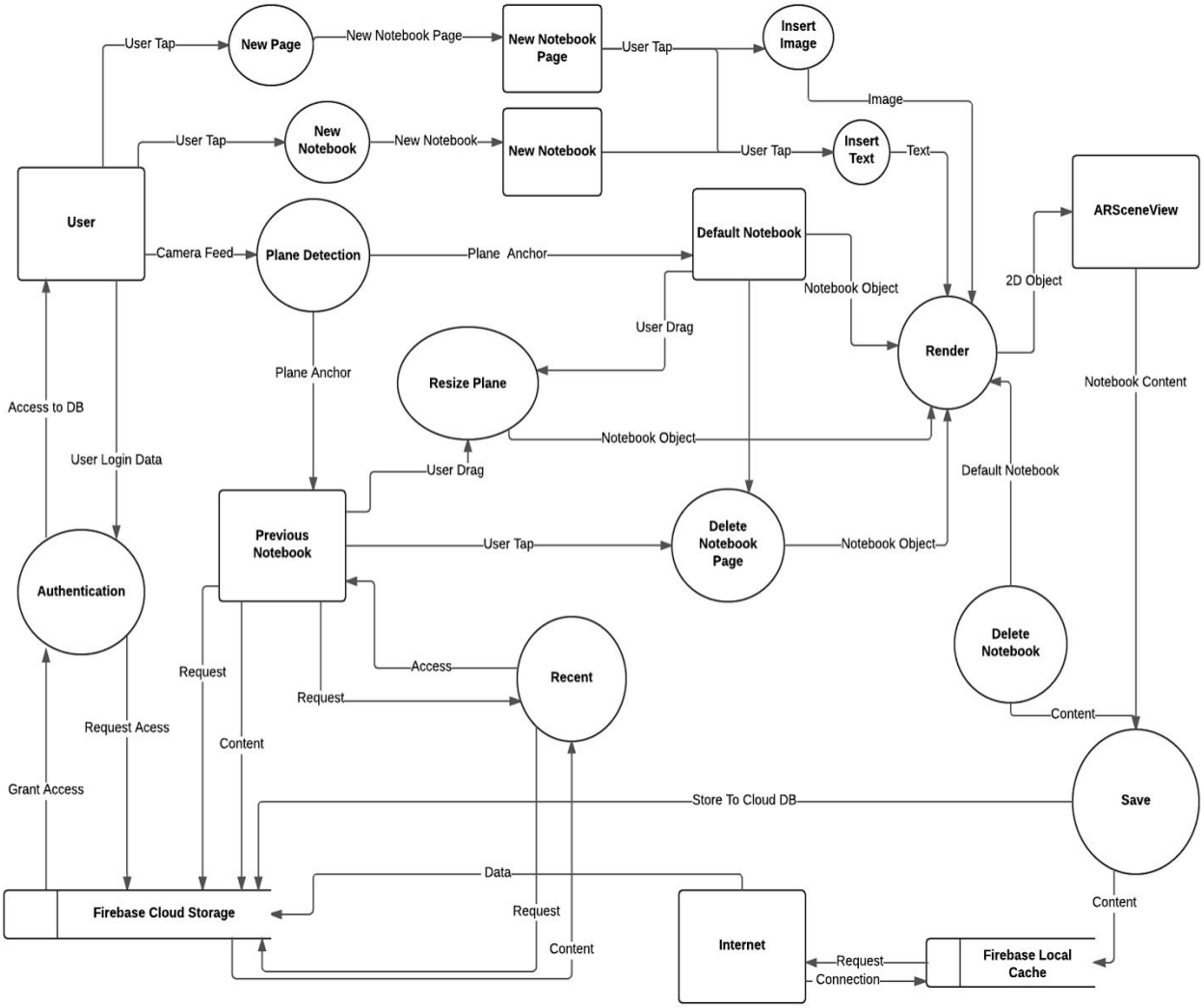

Figure 4: Level 2 DFD

Level 2 Single Process: New Notebook

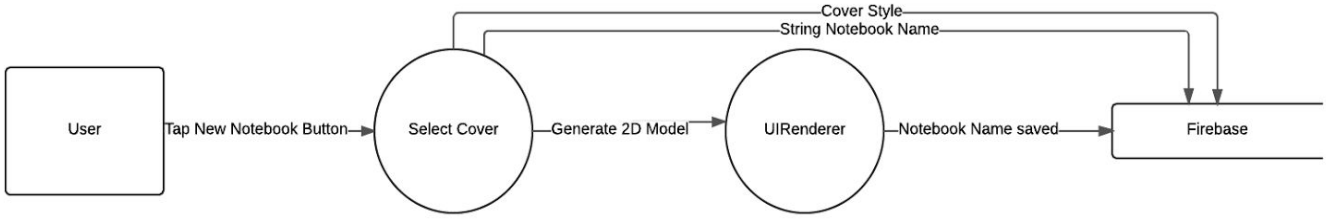

Figure 5: Level 2 DFD for New Notebook

Level 2 Single Process: Retrieve Notebook

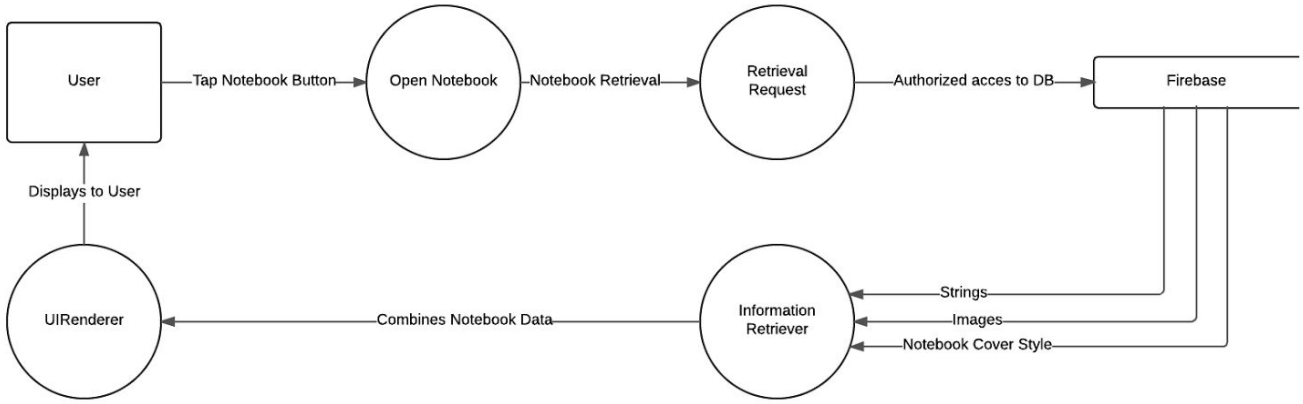

Figure 6: Level 2 DFD for Retrieve Notebook

5.0 User Interface and Wireframes

The User Interface section will display the different interfaces that the user will interact with as well as describing what each portion of the screen moves. The Wireframe Section will show a flow of how the user traverses the application. Specific placement of some buttons required for the app's full functionality have not yet been confirmed. These wireframes may change as we explore ARKit's abilities and determine how much notebook content can comfortably fit while still allowing user interaction on smaller iPhone screens.

5.1 User Interface

NOTE: These wireframes are not final. The core functionality will remain. However, the positioning, the buttons themselves, and additional menu options are subject to change. These wireframes are meant to serve as a guide, not an absolute concept.

ID: UI5.1.1 Authentication

Description: When opening the application for the first time, the user will be prompted to sign in using a Facebook account. The user can skip this step, but they will not be able to retrieve the notebooks they have saved.

Figure 7 : Authentication

ID: UI5.1.2 Dialogue Title

Description: After the authentication, the app will automatically alert with a message telling how to use and proceed within the app.

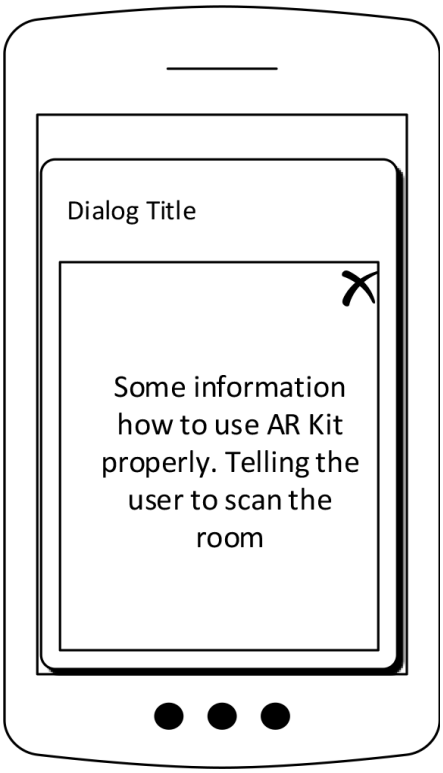

Figure 8: Dialogue Prompt

ID: UI5.1.3 Normal Application View

Description: This is what the application will look like in its normal state. In this screen, the user will be able to see their notebook as an Augmented Reality 3D object. The arrow in the top right corner is an undo button. This button will revert the last change that the user had made in their notebook. The plus button will open a list of menus that will be discussed in more detail in UI5.1.4.

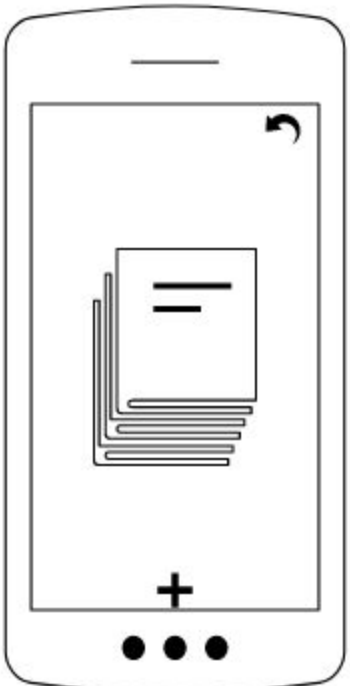

Figure 9: Normal State UI

ID: UI5.1.4 Menu Options

Description: After the user taps the plus button, a menu will appear that allows users to select from several options. There will be an option to create an entirely new notebook, which is discussed more in UI5.1.5. The edit page color option will allow users to change the color of their current notebook page, discussed more in UI5.1.6. The insert option shall allow users to insert a new piece of content on their current notebook page, discussed more in UI5.1.7. The last option is a save option, which saves everything that has been done to the notebook.

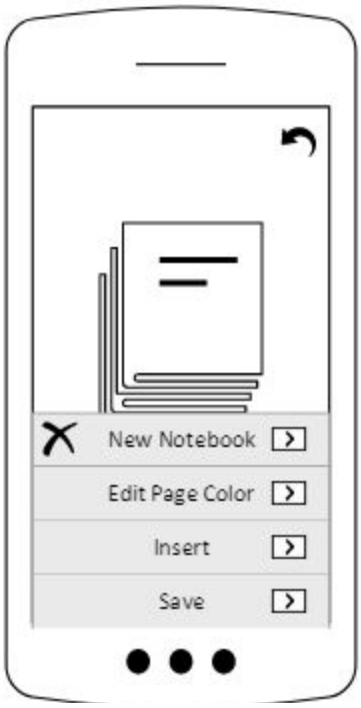

Figure 10: Menu UI

ID: UI5.1.5 New Notebook Button

Description: When the user taps the New Notebook button, they will be asked to choose from a predefined set of notebook cover styles. Although not shown in this screen below, the user will also be asked the name of the notebook as well.

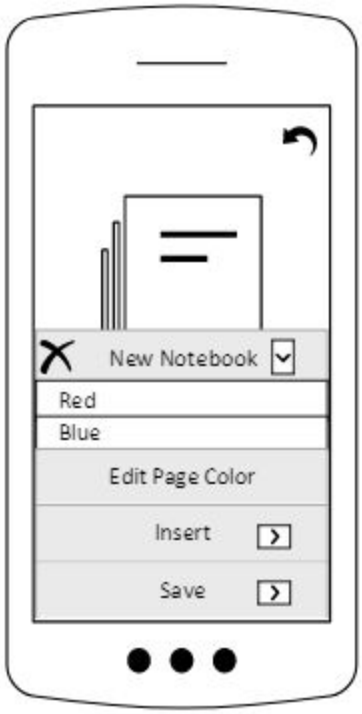

Figure 11: New Notebook Button

ID: UI5.1.6 Edit Page Color

Description: When the user taps the Edit Page Color option, the user will be able to change the color of their current page from a list of predefined colors.

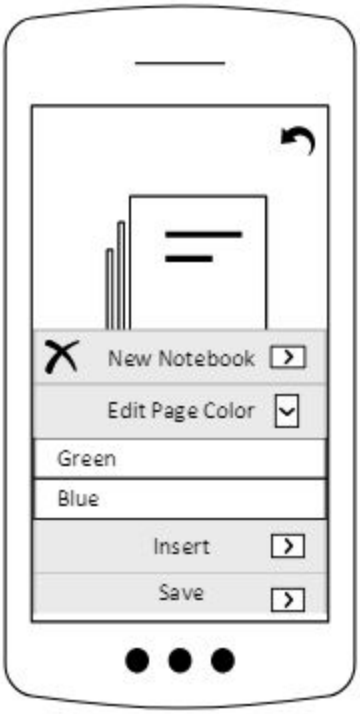

Figure 12:Edit Page Color

ID: UI5.1.7 Insert New Content

Description: When the user taps the Insert option, they will be able to select from three choices: an image from the gallery, text from the keyboard, or pasting from the clipboard.

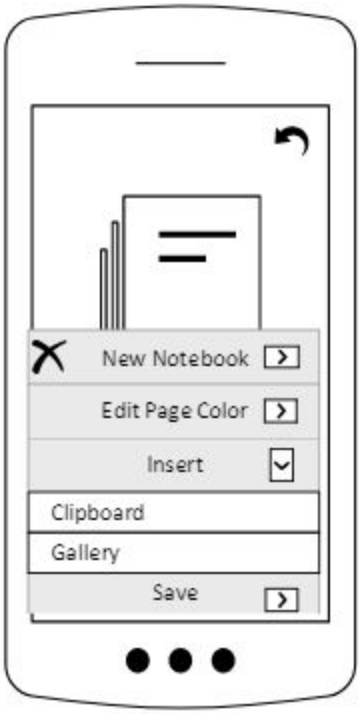

Figure 13: Insert

5.2 Wireframe Flow

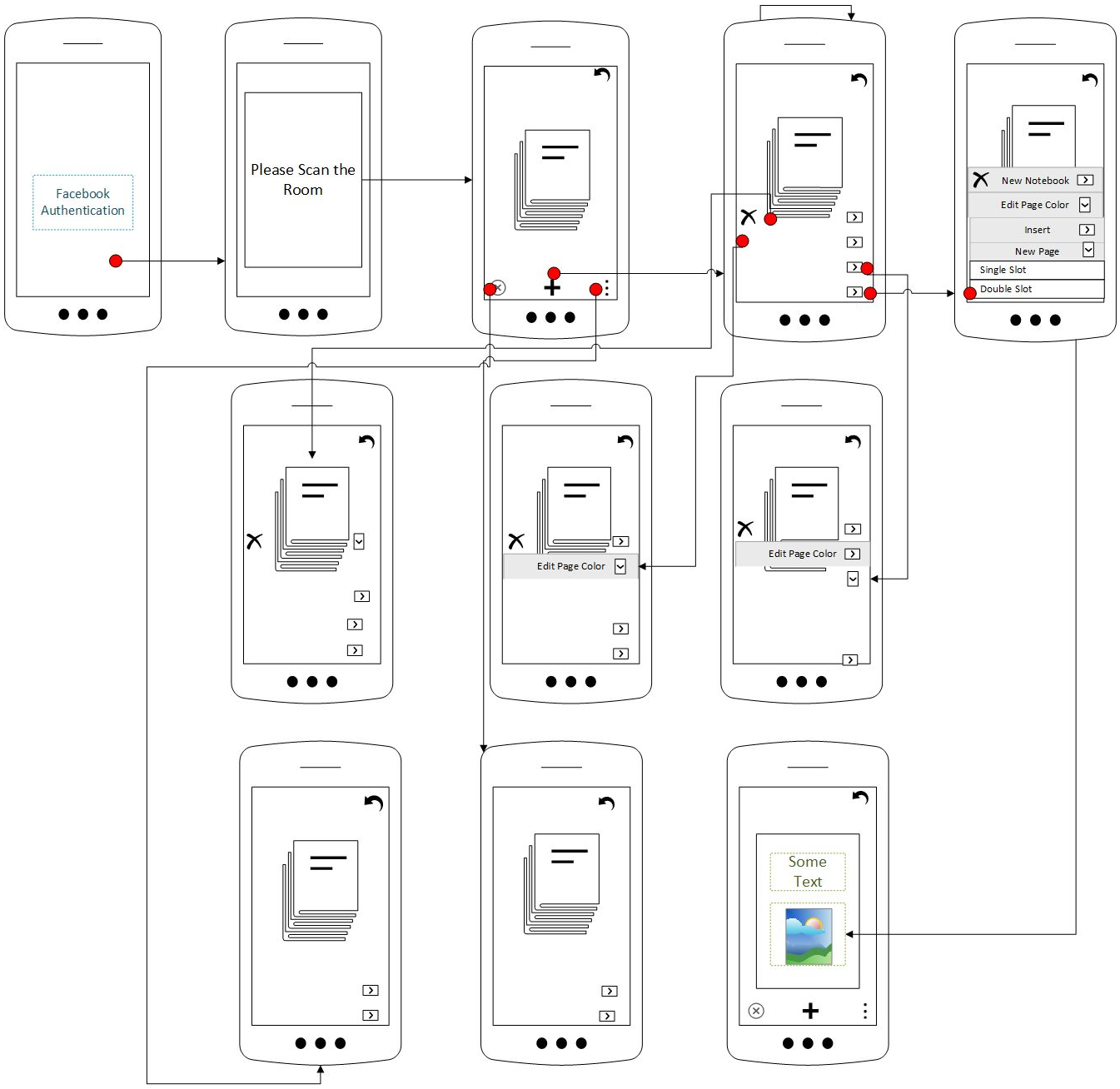

Figure 14: Wireframe Flow

Appendices

6.1 Requirements Traceability Matrix

The traceability matrix is the framework that is used to track each requirement in the project and its related test case through the development of the project.

Requirement ID	Requirement Name	Use Case ID	Test Plan ID
FR3.2.1	**User Authentication**		
FR3.2.2	**Welcome and Instruction Message**		
FR3.2.3	**Create a New Notebook**		
FR3.2.4	**Create a New Notebook Page**		
FR3.2.5	**Insert Text**		
FR3.2.6	**Insert Text from the Clipboard**		
FR3.2.7	**Insert Image from Gallery**		
FR3.2.8	**Turn Notebook Page**		
FR3.2.9	**Save the Notebook**		
FR3.2.10	**Delete a Notebook Page**		
FR3.2.11	**Delete a Notebook**		

FR3.2.12	**Undo**		
FR3.2.13	**Resize Image**		
FR3.2.14	**Retrieve the Notebook**		
FR3.2.15	**Choose from Page Template**		
FR3.2.16	**Sharing Notebooks**		
FR3.2.17	**GPS Notebook Locations**		
FR3.2.18	**Alert the User on Low Battery**		