
System Logical and Physical design

eyeRS Development Team - ITSP200 (Deliverable 3)



Group and Customer Information

Group number	5
Group name	eyeRS Development Team
Members:	Student number: M4DDK8SQ1 Name: Nathan Surname: Shava
	Student number: XQ9X3WV31 Name: Matthew Surname: Van der Bijl
	Student number: MB2015-0785 Name: Emilde Surname: Arsenio
	Student number: Z46WWQH76 Name: Andrea Surname: Cloete
	Student number: MB2015-0219 Name: Sajjaad Surname: Ishmail
Customer	Name: Ndai Mapaso Company: CTI Education Group Industry: Information Technology

Context Page

Introduction	4
1. Information systems design	5
1.1. Logical design	5
1.1.1. EyeRS Entity Relationship Diagram	6
1.1.2. Context & Logical Process Model	7
Context (High Level)	7
Logical Process Model (Low level DFD)	9
1.2 Physical design	10
1.2.1 Investigation of technologies to be applied	10
1.2.2 System testing	13
Introduction	38
1.2.3 System interface design	41
Login Screen	41
Register Screen	43
Forgot PIN Screen	45
Main Menu Screen	47
View Item Screen	49
Item Detail Screen	50
Item Delete Screen	51
Item Share and Trade Screen	52
Edit Item Screen	53
Add Item Screen	55
Add New Category Screen	57
Slideshow Screen	58
Settings Screen	59
Settings > Security Screen	60
Settings > Display Screen	61
Settings > Profile Screen	62
Settings > Sound Screen	63
Setting > Category Management Screen	64
Setting > Category Management > Add Category Screen	65
Setting > Category Management > Edit Category Screen	66
Setting > Category Management > Delete Category Screen	68
Setting > Category Management > Change Sorting Screen	69
Setting >Item Management	70



Setting >Item Management > Add Item Screen	71
Setting >Item Management > Edit Item Screen	73
Setting >Item Management > Delete Item Screen	74
Setting >Item Management > Change Sorting Screen	77
Share and Trade Screen	78
Help and Tips Screen	80
About Screen	81
2. Customer sign-off	82
Appendices	83
Appendix A	83
Bibliography	84



Introduction

This deliverable presents the system logical and physical design document. This document will define what physical and logical design entails and transpire. It also includes the logical data model (ERD), the logical process model (DFD), the testing templates and the test plan. Additionally it includes the system interface designs along with the technologies that have been used in this project.

The purpose of the eyeRS app is to give individuals the power to manage and organise their personal and professional lives with less hassle. The eyeRS app achieves this control by allowing users to freely access and catalog all of their belongings and items that are uploaded to the app. Uploads can occur anywhere. Each belonging is saved by means of an image, title and a short description.

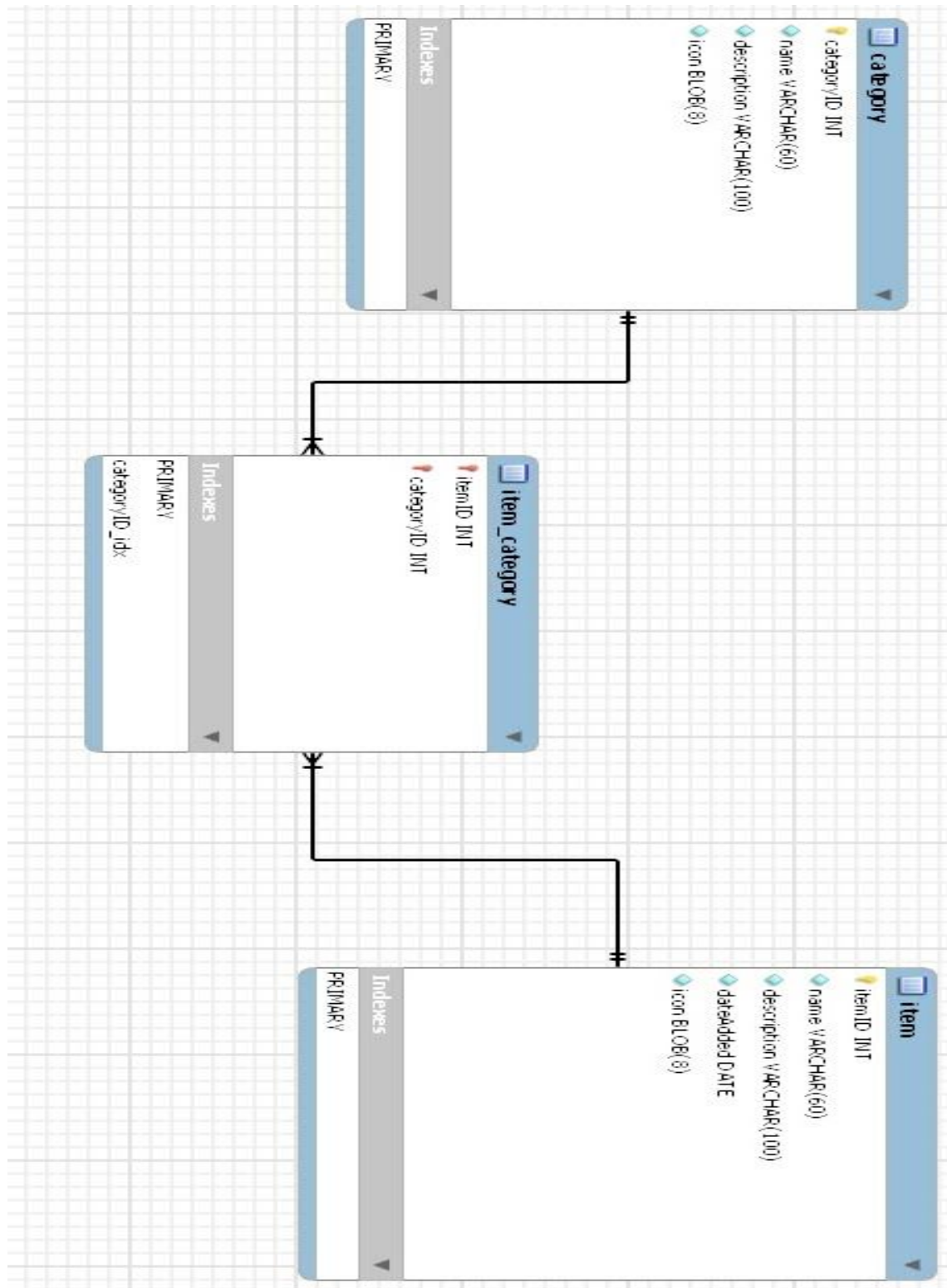


1. Information systems design

1.1. Logical design

Logical design of a system relates to the abstract presentation of the data flows within the system, it is mostly conducted via modelling by using an over-abstract model of the actual system (Satzinger, *et al.*, 2015). Logical design shows how the data that have been given to the system is processed (Satzinger, *et al.*, 2015). This design type is used to document information systems since the logical nature of the system is documented without specifying the detailed tasks of how, where and by whom the information that the system needs, are gathered. Logical designs can be represented by an Entity-Relationship diagram (Mahfuj, 2012).

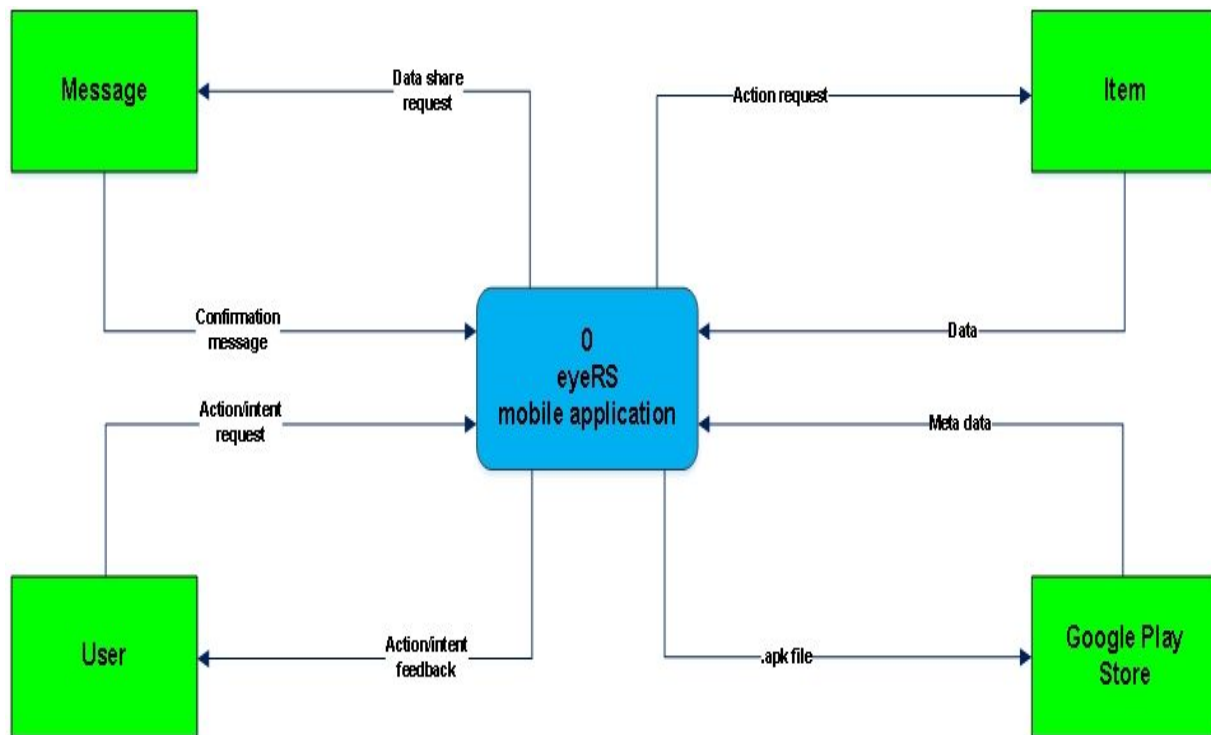
1.1.1. EyeRS Entity Relationship Diagram



1.1.2. Context & Logical Process Model

Context (High Level)

The diagram below indicates the context diagram (high-level DFD) for the eyeRS mobile application.

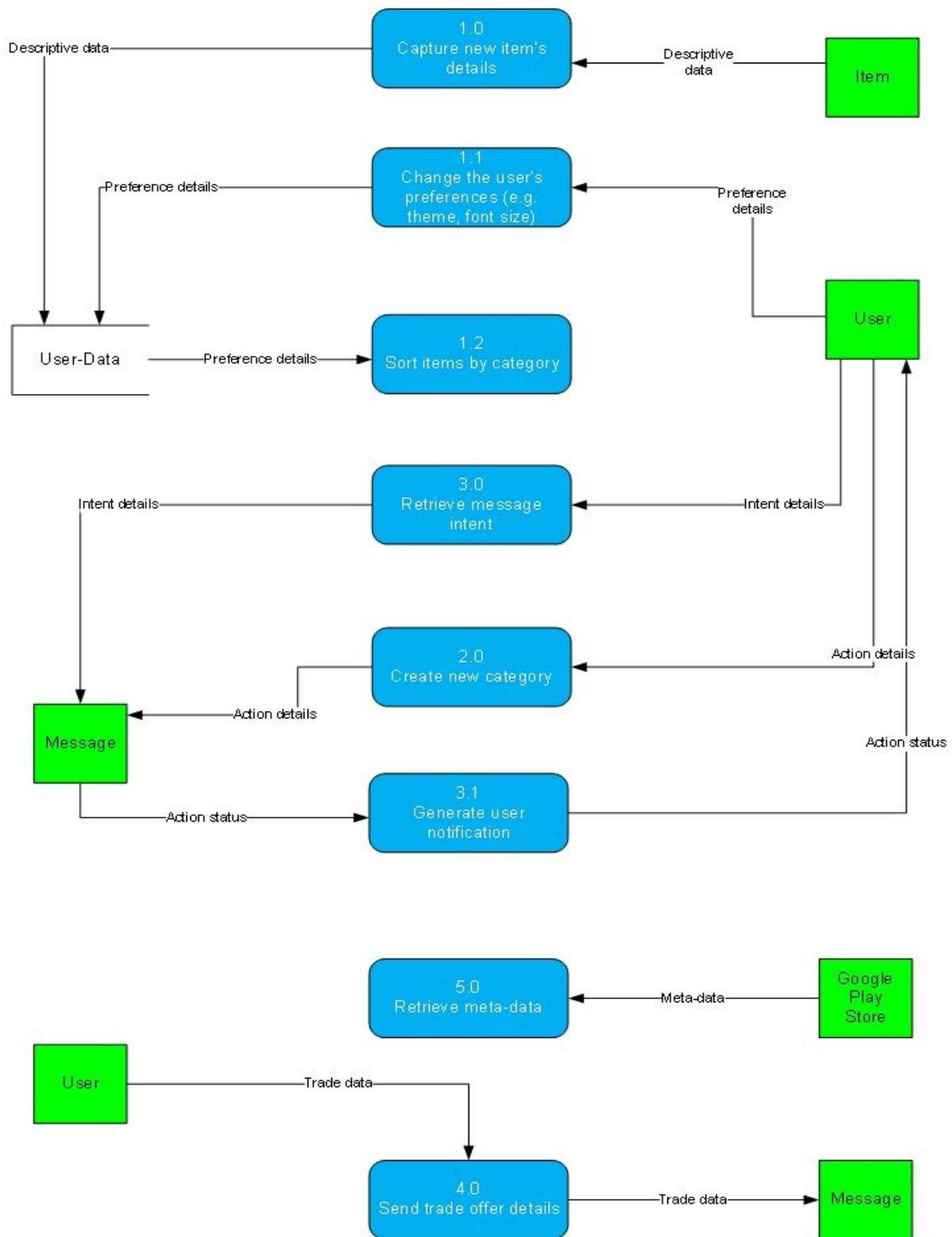


Processes
1.0 Capture the new item's details; 1.1 Get user preferences; 1.2 Sort items by category 2.0 Create new category; 3.0 Retrieve message intent; 3.1 Generate user notification; 4.0 Send trade offer details 5.0 Retrieve metadata

The table below is a summary of the data flows, processes and entities relating to the Data Flow Diagram.

Data Flow	From	To
Intent details	User entity	Retrieve message intent process
Descriptive data	Item entity	Capture new item's details process
Descriptive data	Retrieve item's details process	User-Data data store
Preference details	User entity	Implement the user's preferences process
Preference details	Implement the user's preferences process	User-Data data store
Intent details	Retrieve message intent	Message entity
Action status	Message entity	Generate user notification process
Action status	Generate user notification process	User entity
Meta-data	Google Play Store entity	Retrieve meta-data process
Action details	User entity	Create new category process
Preference details	User-Data data store	Sort items by category process
Trade data	User entity	Validate trade request process
Trade data	Validate trade request process	Message entity

Logical Process Model (Low level DFD)





1.2 Physical design

Physical design is the graphical presentation of the system (Dennis, *et al.*, 2005). It shows the internal and external units. It also shows how the data is flowing within the system (Dennis, *et al.*, 2005). The physical design relates to the input and output process of the system, how data is authenticated, how it is processed and how it is displayed as output (Dennis, *et al.*, 2005). Physical design can be broken down into three sub-tasks. These subtasks of the physical design are:

1. User Interface design;
2. Data design; and
3. Process design.

User interface design focus on how the user gives information to the system and how this data will be presented to the user (Shelly, *et al.*, 2003). Data design is how the data is stored and represented within the system (Shelly, *et al.*, 2003). Process design focuses on how data will move within the system, how and where the data is validated, secured or transformed as it is flowing within the system as well as out of the system (Mahfuj, 2012).

1.2.1 Investigation of technologies to be applied

The operating system that we will be using throughout project development, on the different stages of development from planning until implementation/maintenance will be Windows 7,8 or 10 operating systems.

There are other operating systems on the market, such as: Linux and Mac OS however, Windows will be the preferred operating system of use. On the Windows operating system we will run different technologies to enable successful development of the eyeRS app.

The following technologies are used throughout the development of the eyeRS project:

1. Justinmind;
2. Android studio;
3. Microsoft word;
4. Github;
5. Google Drive;
6. Java Runtime Environment (JRE);
7. Java Development Kit (JDK);
8. Android Software Development Kit Manager;
9. Android virtual Device Manager; and
10. SQLite.



The above mentioned technologies are briefly discussed below:

Justinmind is a system that allows users to create system and application prototypes, supporting many devices.(Justinmind, 2014) It uses a drag and drop feature and assign events to the objects (Justinmind, 2014). It will assist us in creating user interfaces and allow us to gain feedback from the client of specific design preferences.

Android studio is a development environment for the Android platform that we will use to develop, debug and deploy the eyeRS app (Android Developers, 2017). It provides the best tools for building high quality apps for any android device (Android Developers, 2017).

Microsoft word is a graphical word processing system that users can create documents with (WebAIM, 2016). We are using it for the project documentation . The documentation of the project is important for many reasons such as:

1. A reference for other projects;
2. For system upgrade or enhancement;
3. Project troubleshooting and;
4. Providing an understanding of the system concept behind its functionality (Shelly, et al., 2003).


Github is a repository hosting service (Finley, 2012). It will allow the team to collaborate on development tasks via its platform, in a secure and productive way (Finley, 2012). It allows project team members to work on the project tasks from anywhere regardless of location (Github, 2017).

Google Drive will provide access to files anywhere in a secure way, working as a cloud storage to backup images, videos and documents (Google, 2017). It also enable the project team to utilize Google Docs, which is an online, word processing program to compile documents allow the team to work on the same document simultaneously (Google, 2017).

Java Runtime Environment is a software package that consists of the necessary requirements to execute a Java program (Oracle, 2017). The eyeRS team will make use of this technology (which comes as part of the Android Studio SDK) to create the app's functionality (Android Developers, 2017).

Java Development Kit is will be used to develop java programs as it also comprises of the Java Runtime Environment in the kit (Oracle, 2017).

Android Software Development Kit Manager is a set of development tools which are embedded in the Android Studio development environment which will be used to debug the eyeRS app (Android Developers, 2017).



Android virtual Device Manager is a program that will enable the development team to debug their apps on a virtual machine with specific requirements such as: hardware, storage space, screen resolutions and allowing the development team to target various Application Programming Interface (API) levels (Android Developers, 2017).

SQLite is a database management system based on the structured query language. (Oracle, 2017). It is a technology used by the Android system to create databases to be used for storage by the application (Android Developers, 2017).

1.2.2 System testing

Testing types

Software testing is the ongoing process of ensuring that a program not only is function but also fulfils all set criteria for the program (IEEE, 1990). According to IEEE (1990) and Williams (2006) there are various primary types of programing tests that are should be performed on a program before it can be labeled as acceptable, these tests include:

1. Walkthroughs;
2. Trace table
3. Black box/data coverage; and
4. White box/Code coverage.

Bertolino (2001) suggested that software testing be conducted throughout the development process.

Walkthroughs are a fairly common and cost effective method of software testing. Walkthroughs are done by manually tracking a program's data and output on pen and paper. Walkthroughs are frequently used to facilitate discussion to what extent a program has met its design requirements. Trace tables are similar to walkthough however they are focused on testing the logic of an algorithm rather than the program as a whole (ref). According to ref trace tables are good for highlighting small non obvious logic errors with an a program.

Black-box testing, also known data coverage, is when a subprogram verifies a set of known valid inputs are tested against possible input values (Williams, 2006). According Williams (2006) to black-box testing is used to verify that a program can process possible values and meets the proposed requirements.

White-box testing, also known as code coverage, is when the execution of statements, or groups of statements, are tested to ensure that they execute as expected (Williams, 2006). Williams (2006) stated White-box testing is used to ensure that all statements execute as expected.

Softwaretestingclass (2012) states that system testing tests the behaviour of the system based on risks, specifications and requirements which include functional and nonfunctional requirements that were established by the developers. A fully integrated system should be tested. The final testing done is the Systems Testing to verify that all specifications and requirements have been met.

Testing templates

Test Case Number: 1 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Test the registration for eyeRS	Test Case Name: Register new user Subsystem: Register Design Date: Execution Date:
--	---

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click "Register" button	The app displays a screen with registration details		
2	Enter a username	The app displays the username in the text area		
3	Enter an email address	The app displays the email address in the text area		
4	Enter a pin "*****"	The app asks the user to create their PIN		
5	Re-enter the pin "*****"	The app verifies that the pins match		
6	Select a security question	The app opens a drop-down list		
7	Enter security response (Answer to security question).	The app accepts the input		
8	Click the "Register" button	The system registers the user with the database, a Registered notification appears and returns to the "Login" screen		

Test Case Number: 2 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Allows the user the opportunity to retrieve a forgotten pin	Test Case Name: Forgot Pin Subsystem: Main Menu Design Date: Execution Date:
--	---

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on the red "Forgot Pin" text	New screen appears that display security question and text field for the security question answer.		
2	Enter a new PIN in the "Create New PIN" text field	The app asks the user to create their new PIN		
3	Verify the new PIN in the "Verify PIN" text field	The app verifies that the new PINs match		
4	Enter your registered security question	The user selects the security question that they registered to allow them to reset their PIN from the drop down box.		
5	Enter your registered security response	Edit text field accepts the user's input which should match the one that was captured during registration		
6	Click the "OK " button	The app compares the answer with one in the database if matched a pop-up is displayed notifying the user of successful PIN change and redirects to the Login screen. If they do not match then a pop-up screen will display requesting them to re-enter their security details.		

Test Case Number: 3 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Log in to use the eyeRS application	Test Case Name: Login Subsystem: Main Menu Design Date: Execution Date:
--	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Enter pin	PIN appears as hidden password input		
2	Click the "Login" button	If correct the system redirects to the main menu screen. If not the app will notify the user of the incorrect PIN, and the PIN can be re-entered. The app redirects to the Main menu upon successful login.		

Test Case Number: 4 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: View items from the main menu	Test Case Name: Viewing Items Subsystem: Main Menu Design Date: Execution Date:
--	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click a category from the main menu	A new Screen with a list of all the items (Item's image, item's title and item's description) will appear (e.g. My Books).		
2	Click an item to view	Item details screen will appear that provide more detail of the item.		

Test Case Number: 5 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: View and edit items from the user's uploads	Test Case Name: Edit Item Subsystem: Item Detail Design Date: Execution Date:
--	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	In the Item Detail screen click the "Edit" button.	The Edit item screen, with edit capabilities will appear.		
2	Change item image by clicking on the image.	Notification appears asking for an upload of the image via the camera or the gallery.		
3	Upload Image via camera	Camera opens - follow steps used to take a photo – photo appears as the image of the item in the eyeRS app. (Note: Changes are not yet saved).		
4	Upload via gallery	Gallery opens- select the appropriate image - image will appear as the image of the item in the eyeRS app. (Note: Changes are not yet saved).		
5	Select the category drop box	A list of all the categories fill appear		
6	Select the appropriate category from the drop box list	The selected category will appear. (Note: Changes are not yet saved).		
7	Change the title of the item in the item text field	Title will appear as the changed value. (Note Changes are not yet saved).		
8	Change the description of the item in the description text field	Description will appear as the changed value. (Note: Changes are not yet saved).		
10	Click the "Save" button.	The changes that were made will now be saved in the database. Notification of saved appears		
11	Click on the "OK" button on the notification	The item detail screen appears		

Test Case Number: 6 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Tests the Enter and Exit button	Test Case Name: Enter and Exit Edit Item Subsystem: Item Detail Design Date: Execution Date:
--	---

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click "Edit " button	Edit item screen, with edit abilities will appear.		
2	Click the "Back Arrow" (←)	The Item detail screen will appear, without any changes to it.		

Test Case Number: 7 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Share items with other eyeRS application user's	Test Case Name Share Item Subsystem: Item Detail Design Date: Execution Date:
--	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on the more icon.	A pop up with more options (Delete, Share and Trade) will appear in the top right corner of the screen.		

2	Click on the "Share" option in the pop up screen	A pop up will appear that shows all the supported sharing methods.		
3	Select the the share media	Share method will open – follow steps as normal for the chosen method		

***Note: Repeat process for each sharing method**

Test Case Number: 8 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: : Trade items with other eyeRS application user's	Test Case Name Trade Item Subsystem: Item Detail Design Date: Execution Date:
--	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on the more icon.	A pop up with more options (Delete, Share and Trade) will appear in the top right corner of the screen.		
2	Click in the "Trade" option in the pop up screen	A pop up will appear that shows all the supported sharing methods.		
3	Select the the share media	Share method will open – follow steps as normal for the chosen method		

***Note: Repeat process for each sharing method**

Test Case Number: 9 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Deletes item from a category	Test Case Name Delete Item Subsystem: Item Detail Design Date: Execution Date:
---	---

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on the more icon.	A pop up with more options (Delete, Share and Trade) will appear in the top right corner of the screen.		
2	Click in the "Delete" option in the pop up screen	Notification that asks for confirmation appears.		
3	Click the "No" button in the Delete conform notification	Delete confirm notification will disappear and the item detail screen will appear.		
4	Click in the "Delete" option in the pop up screen	Notification that asks for confirmation appears.		
5	Click the "Yes" button in the Delete conform notification	Successful deletion notification appears.		
6	Press "OK " button on the delete notification	Item is deleted from the database. My Category screen will appear that contain a list of all the items in that specific category that were chosen.		

Test Case Number: 10 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Searches for items in the database	Test Case Name: Search Item Subsystem: Main Menu Design Date: Execution Date:
--	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on the search icon	Search bar will pop up and user's keyboard will appear.		
2	Search any item	Keyboard will disappear and search results will appear.		

Test Case Number: 11 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Exit from 'My Category' screen	Test Case Name: Exit From My Category Screen Subsystem: My Category Design Date: Execution Date:
--	---

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on the back arrow (←)	Main screen will appear.		

Test Case Number: 12 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Test the registration for eyeRS	Test Case Name: Search Category Subsystem: My Category Design Date: Execution Date:
---	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on the search icon	Search bar will pop up and user's keyboard will appear.		
2	Search any item	Keyboard will disappear and search results will appear.		

Test Case Number: 13 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Displays side menu	Test Case Name: View Side Menu Subsystem: My Category Design Date: Execution Date:
--	---

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on the "More" icon in the main menu	A side menu with more options will appear.		

Test Case Number: 14 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Adds an item to the users database	Test Case Name: Add Item Subsystem: My Category Design Date: Execution Date:
--	---

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on “Add new item” in the side menu	An Add Item screen will appear.		
2	Click on the “Camera” icon	Notification appears asking for an upload of the image via the camera or the gallery.		
3	Upload Image via camera	Camera opens - follow steps used to take a photo – photo appears as the image of the item in the eyeRS app. (Note: Changes not yet saved).		
4	Upload via gallery	Gallery open s- select the appropriate image - image will appear as the image of the item in the eyeRS app. (Note: Changes not yet saved).		
5	Select the category drop box	A list of the categories fill appears.		
6	Select the appropriate category from the drop box list	The selected category will appear. (Note: Changes not yet saved).		
7	Enter the title of the item in the item text field	Title will appear in the text field. (Note: Not yet saved).		
8	Enter the description of the item in the description text field	Description will appear in the description text field. (Note Change is not yet saved).		
9	Click the “Add” button.	Item is added in the database.		

		Notification of saved appears for a short while and the item detail screen appears after the saved message appeared.		
10	Click on the "OK" button on the notification	The main menu screen appears.		

Test Case Number: 15 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Exits add item screen	Test Case Name: Exit Add Item Subsystem: My Category Design Date: Execution Date:
---	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on "Add new item" in the side menu	An Add Item screen will appear.		
2	Click on the back arrow. (←)	Main menu appears.		

Test Case Number: 16 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Adds a category to the user's database	Test Case Name: Add Category Subsystem: Main menu Design Date: Execution Date:
--	---

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on "Add new category" in the side menu	An Add Category screen will appear.		
2	Click on the "Camera" icon	Notification appears asking for an upload of the image via the camera or the gallery.		
3	Upload Image via camera	Camera opens - follow steps used to take a photo – photo appears as the image of the item in the eyeRS app. (Note: Changes not yet saved).		
4	Upload via gallery	Gallery opens - select the appropriate image - image will appear as the image of the item in the eyeRS app. (Note: Changes not yet saved).		
5	Enter the title of the item in the item text field	Title will appear in the text field. (Note: Changes not yet saved).		
6	Enter the description of the item in the description text field	Description will appear in the description text field. (Note: Changes not yet saved).		
7	Click the "Add" button.	Item is added in the database. Notification of saved changes appears.		
8	Click on the "OK" button on the notification	The main menu screen appears.		

Test Case Number: 17 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Exits add category screen	Test Case Name: Exit Add Category Subsystem: Main menu Design Date: Execution Date:
---	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on "Add new category" in the side menu	An Add Category screen will appear.		
2	Click on the back arrow. (←)	Main menu appears.		

Test Case Number: 18 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: View images from the user catalogue	Test Case Name: Slideshow Subsystem: Main menu Design Date: Execution Date:
---	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on "Slideshow" in the side menu	A slideshow screen will appear and the images in the catalog will appear as a gallery view		
2	Click on the back arrow. (←)	Main menu appears.		

Test Case Number: 19 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Displays help and tips for the functionality of eyeRS	Test Case Name: Help and Tips Subsystem: Main menu Design Date: Execution Date:
---	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on "Help and Tips" in the side menu	Help and Tips screen will appear.		
2	Click on the video.	A tutorial video will play.		
3	Type a feedback message.	Message will appear in the text field.		
4	Click the "send" icon.	Notification appears validating success.		
5	Click on "OK" in the notification.	Help and Tips screen appears.		
6	Click the back arrow. (←)	Main menu appears.		

Test Case Number: 20 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Gives a brief history on the development team for eyeRS and the product	Test Case Name: About Subsystem: Main menu Design Date: Execution Date:
---	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on "About" in the side menu	An About screen will appear.		
2	Click the back arrow. (←)	Main menu appears.		

Test Case Number: 21
System: eyeRS
Designed by: eyeRS Development Team
Executed by:
Description: Shares a standard or custom user category and all content with other user's

Test Case Name: Share
Subsystem: Main menu
Design Date:
Execution Date:

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on "Share" in the side menu	A select category screen will appear.		
2	Click on appropriate category.	A Select Item screen will appear.		
3	Select an item to share.	A pop up will appear that shows all the supported sharing methods.		
4	Select the appropriate sharing method	Share method will open – follow steps as normal for the chosen method.		

***Note: Repeat for each sharing method.**

Test Case Number: 22 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Searches for items by (name, category, details)	Test Case Name: Search Item Subsystem: Main menu Design Date: Execution Date:
---	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on the search icon	Search bar will pop up and user's keyboard will appear.		
2	Search any item	Keyboard will disappear and search results will appear.		

Test Case Number: 23 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Exits share screen	Test Case Name: Exit Share Subsystem: Main menu Design Date: Execution Date:
--	---

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click the back arrow. (←)	The select category screen will appear.		
2	Click the back arrow. (←)	Main menu will appear.		

Test Case Number: 24 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Trade category and all content with other user's	Test Case Name: Trade Subsystem: Main menu Design Date: Execution Date:
--	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on "Trade" in the side menu	A select category screen will appear.		
2	Click on appropriate category.	A Select Item screen will appear.		
3	Select an item to share trade details for.	A pop up will appear that shows all the supported sharing methods.		
4	Select the appropriate sharing method	Share method will open – follow steps as normal for the chosen method.		

Test Case Number: 25 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Searches for items by (name, category, details)	Test Case Name: Search Item Subsystem: Main menu Design Date: Execution Date:
---	--

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click on the search icon	Search bar will pop up and user's keyboard will appear.		
2	Search any item	Keyboard will disappear and search results will appear.		

Test Case Number: 26 System: eyeRS Designed by: eyeRS Development Team Executed by: Description: Exits share screen	Test Case Name: Exit Share Subsystem: Main menu Design Date: Execution Date:
--	---

Test Number	Action	Expected Response	Pass/Fail	Comments
1	Click the back arrow. (←)	The select category screen will appear.		
2	Click the back arrow. (←)	Main menu will appear.		

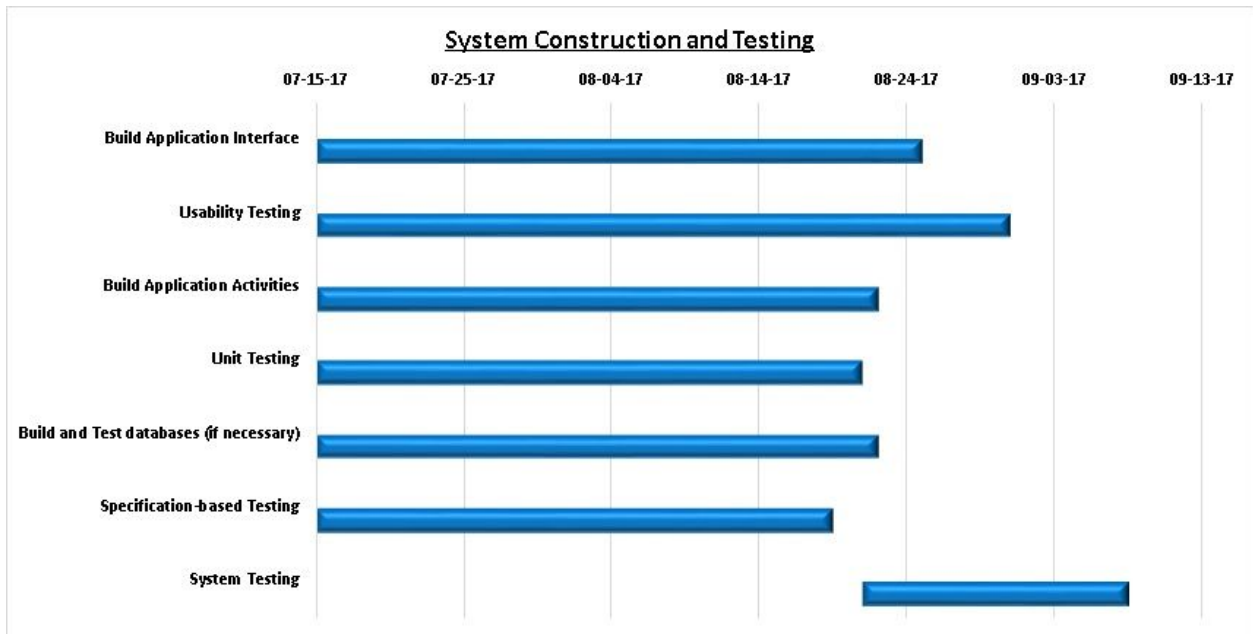
Test plan

Introduction

Our project is a mobile app, which is intended for operation on the Android OS platform. Hardware resources intended for testing will comprise of personal computers for compiling and emulation, together with Android smart-phones and tablets for testing on real devices. Software being used includes emulators, Javadocs, GitHub and the Android Developer console for additional resources. The team is currently using Android Studio which has various mobile and tablet device emulators that will utilize a computer's different Hyper-V features to run the emulators. All resources are available for each member of the eyeRS development team. Testing and debugging could take up to a total of approximately 48 hours.

Test Plan - eyeRS Mobile App			
Test type	Test date	Team members	Description
Unit (White box)	30/08/2017	Mr. Matthew Van Der Bijl, Mr. Nathan Shava	All the activities and/or fragments' code statements will be tested to ensure that they execute as expected.
Unit (Black box)	31/08/2017	Mr. Matthew Van Der Bijl, Mr. Nathan Shava	The app's activities and/or fragments will be tested to verify that each activity can process certain inputs and meet proposed requirements.
Integration	06/09/2017	Mrs. Ndai Makhurane (Customer), Mr. Matthew Van Der Bijl, Mr. Nathan Shava, Ms. Andrea Cloete, Mr. Sajjaad Ishmail, Mr. Emilde Arsenio	Here the entire app's activities and/or fragments will be integrated to verify whether they interact all as expected.
System & Stress	07/09/2017	Mrs. Ndai Makhurane (Customer), Mr.	The performance of the system will be tested at this phase

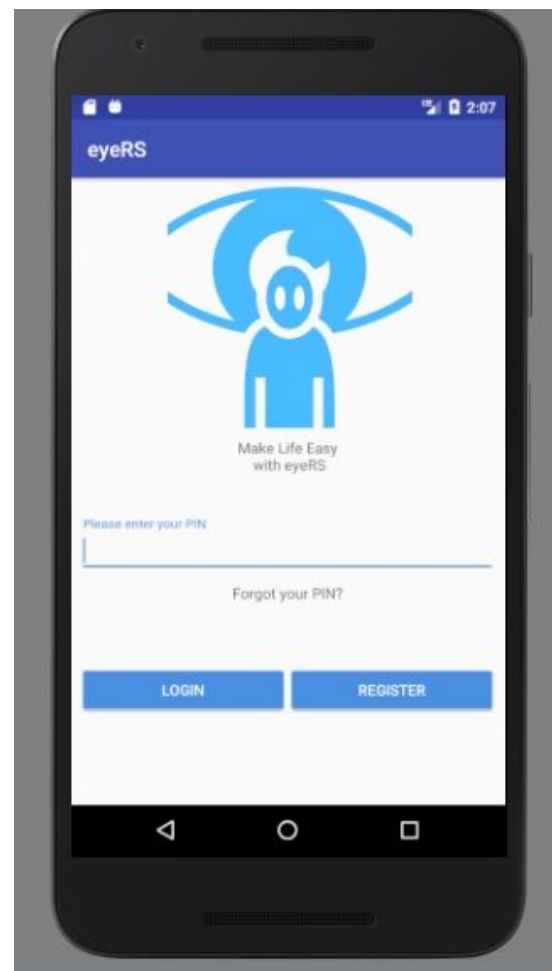
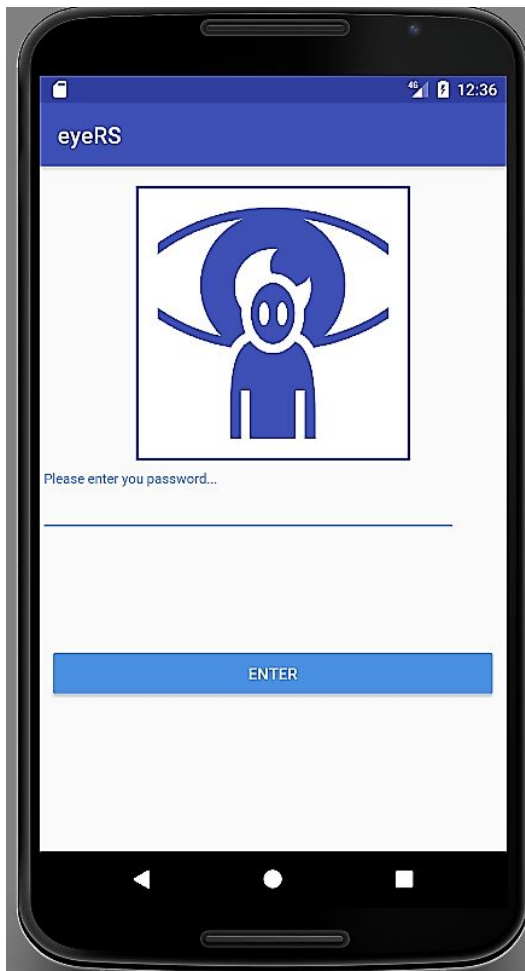
		Matthew Van Der Bijl, Mr. Nathan Shava, Ms. Andrea Cloete, Mr. Sajjaad Ishmail, Mr. Emilde Arsenio	to find out whether the app's responsiveness, for example, is at an acceptable level.
Entire App Walkthrough (Usability)	07/09/2017	Mrs. Ndai Makhurane (Customer), Mr. Matthew Van Der Bijl, Mr. Nathan Shava, Ms. Andrea Cloete, Mr. Sajjaad Ishmail, Mr. Emilde Arsenio	The entire app will be tested simulating the entire user experiences to debug for any other errors not detected during the other phases.
Acceptance	13/09/2017	Mrs. Ndai Makhurane (Customer), Mr. Matthew Van Der Bijl, Mr. Nathan Shava, Ms. Andrea Cloete, Mr. Sajjaad Ishmail, Mr. Emilde Arsenio	The proposed customer will test the app to verify if it has meet all the stipulated requirements.
Review & Corrections	14/09/2017	Mr. Matthew Van Der Bijl, Mr. Nathan Shava	Time allocated for corrections based on the review process.

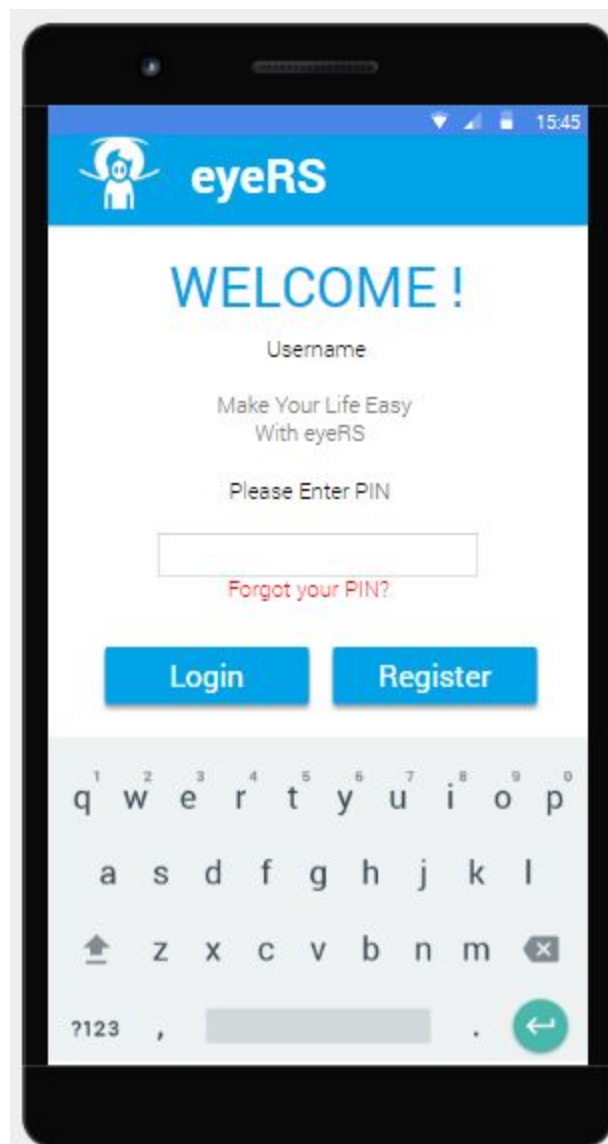


1.2.3 System interface design

Login Screen

This screen will allow the user to gain access to the app by only supplying a PIN as there can only be one sole user of the app per device.





Register Screen

The image displays two mobile app screens for the 'eyeRS' application, showing the registration process. Both screens have a blue header with the 'eyeRS' logo and a status bar at the top showing the time as 15:45.

Left Screenshot (Registration Form):

- Header: eyeRS
- Section: WELCOME !
- Text: * These fields are required
- Form Fields:
 - Enter Username*
 - Enter e-mail address*
 - Create your new PIN*
 - Verify PIN*
 - Please select a security question* (dropdown menu)
 - Security Response*
- Button: Register

Right Screenshot (Registration Successful):

- Header: eyeRS
- Section: WELCOME !
- Text: * These fields are required
- Form Fields (faded):
 - Enter Username*
 - Enter e-mail address*
 - Verify PIN*
 - Please select a security question* (dropdown menu)
 - Answer*
- Notification Overlay:
 - Notification
 - Register successful
- Button: Register

eyeRS

Welcome!

*These fields are required

Enter Username

Enter Email

Please enter your new PIN

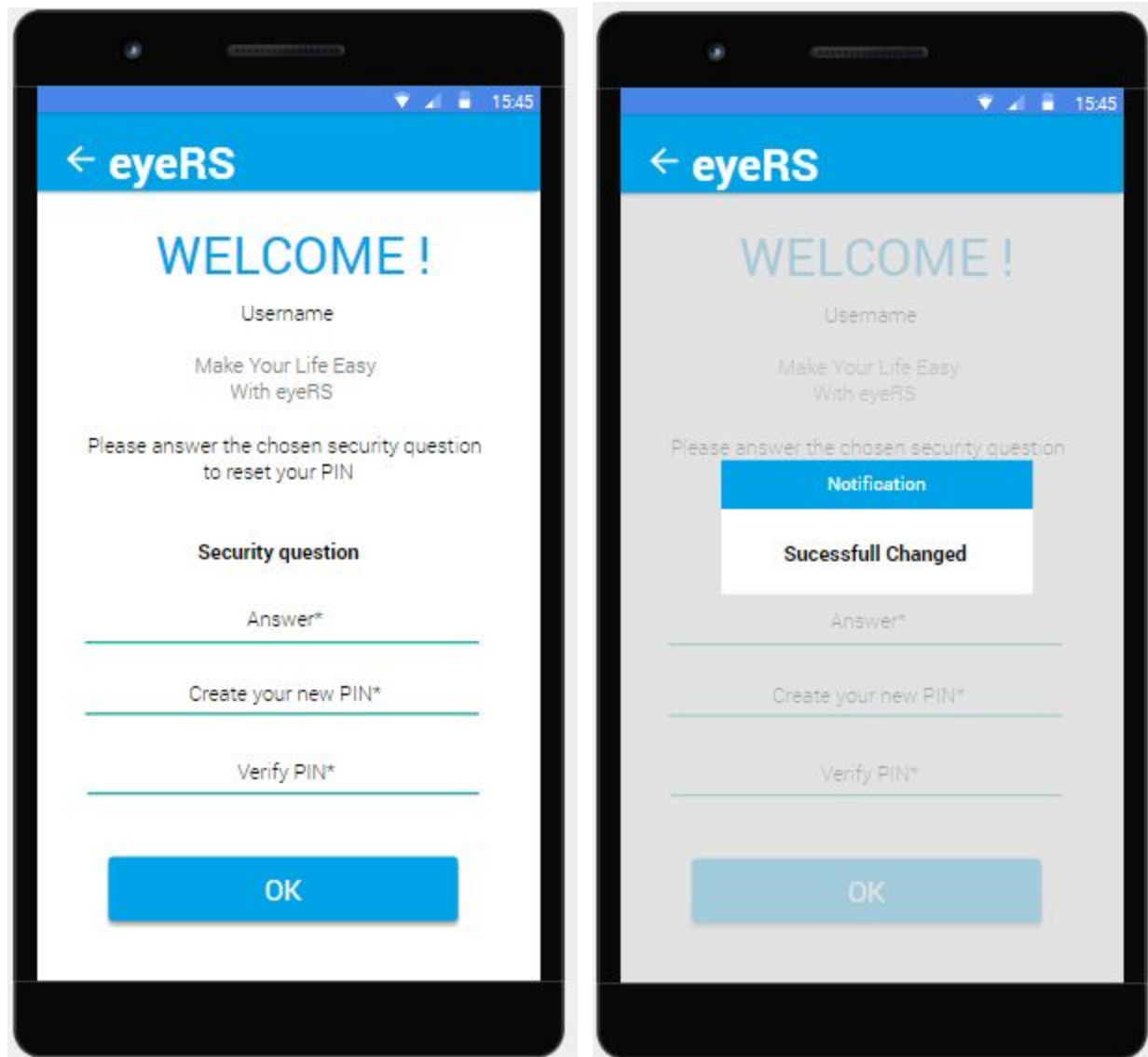
Please confirm your new PIN

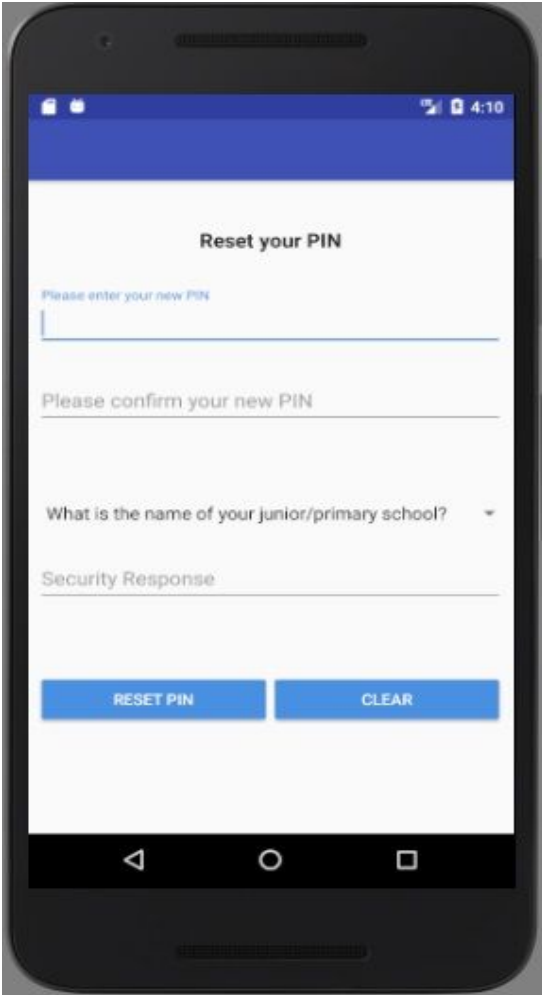
What is the name of your junior/primary school? ▾

Security Response

REGISTER

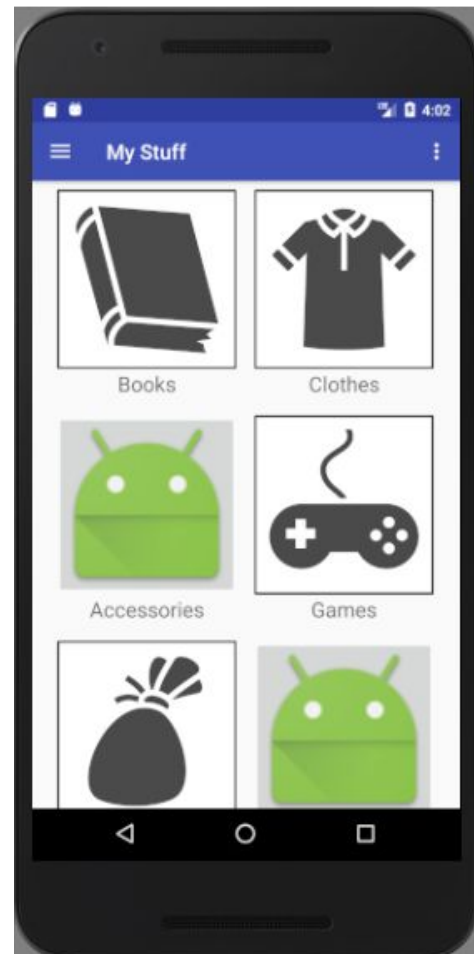
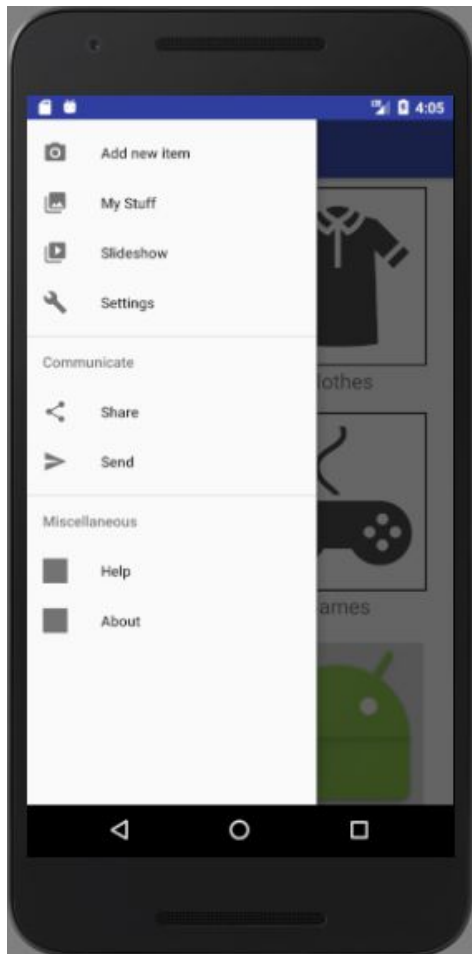
Forgot PIN Screen



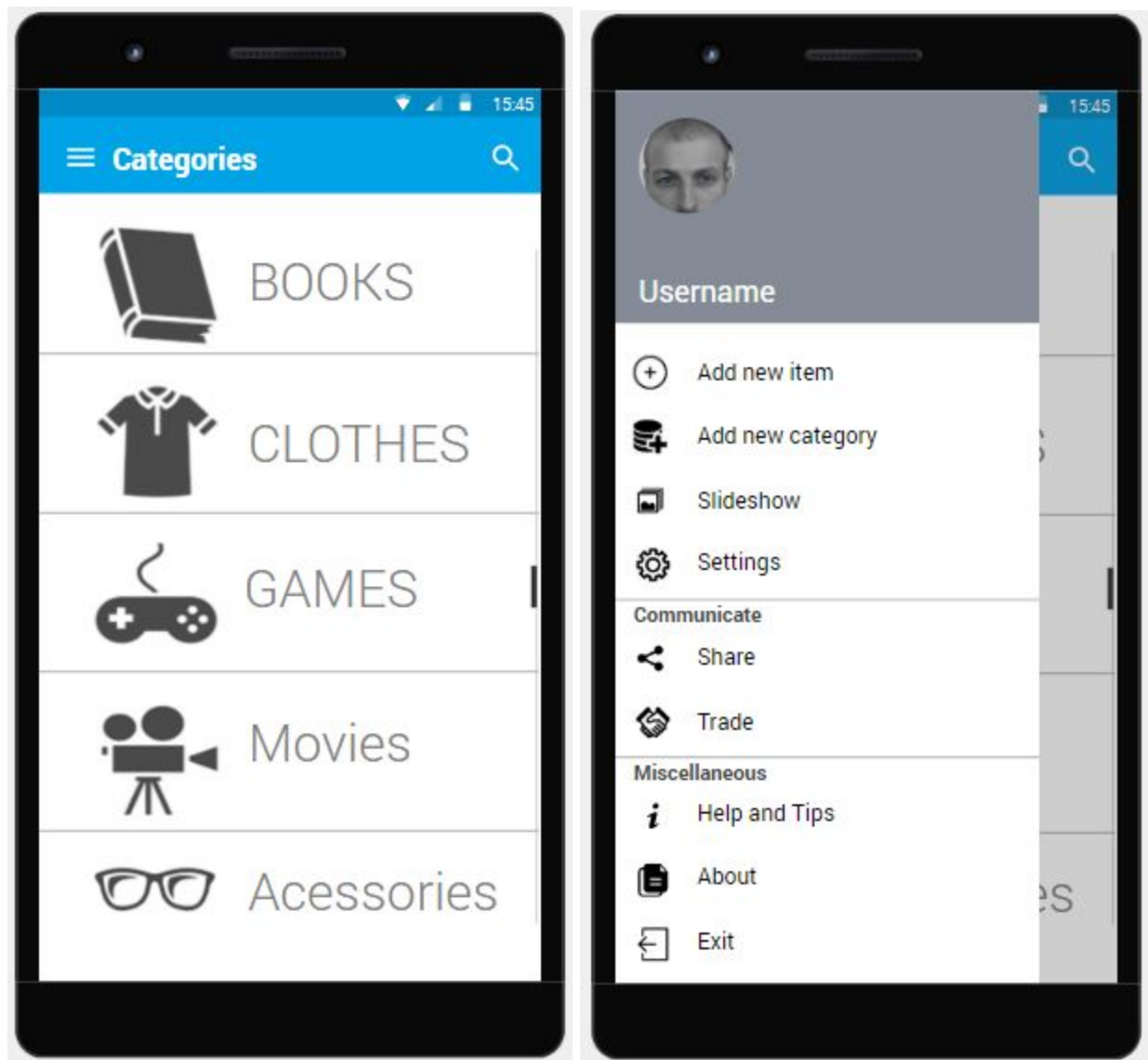


Main Menu Screen

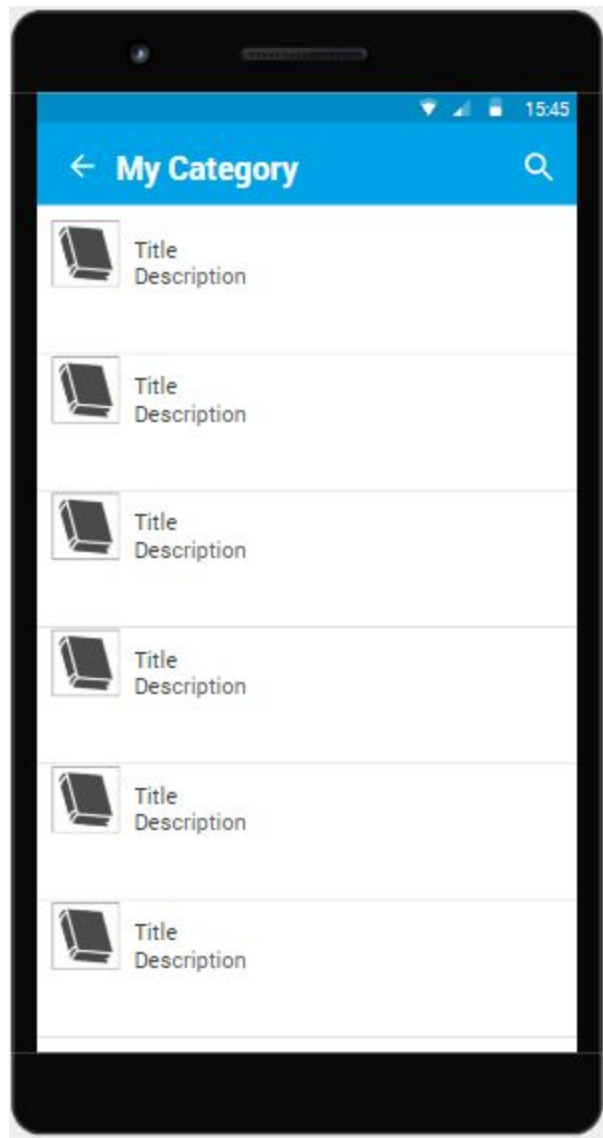
This screen will consist of a Navigation drawer in which the user can access by swiping right from the left edge of the screen or by clicking on the drawer button found on the top left corner of the screen on the action bar.



The My Stuff screen also features a graphical UI consisting of clickable icons representing the app's catalog categories.



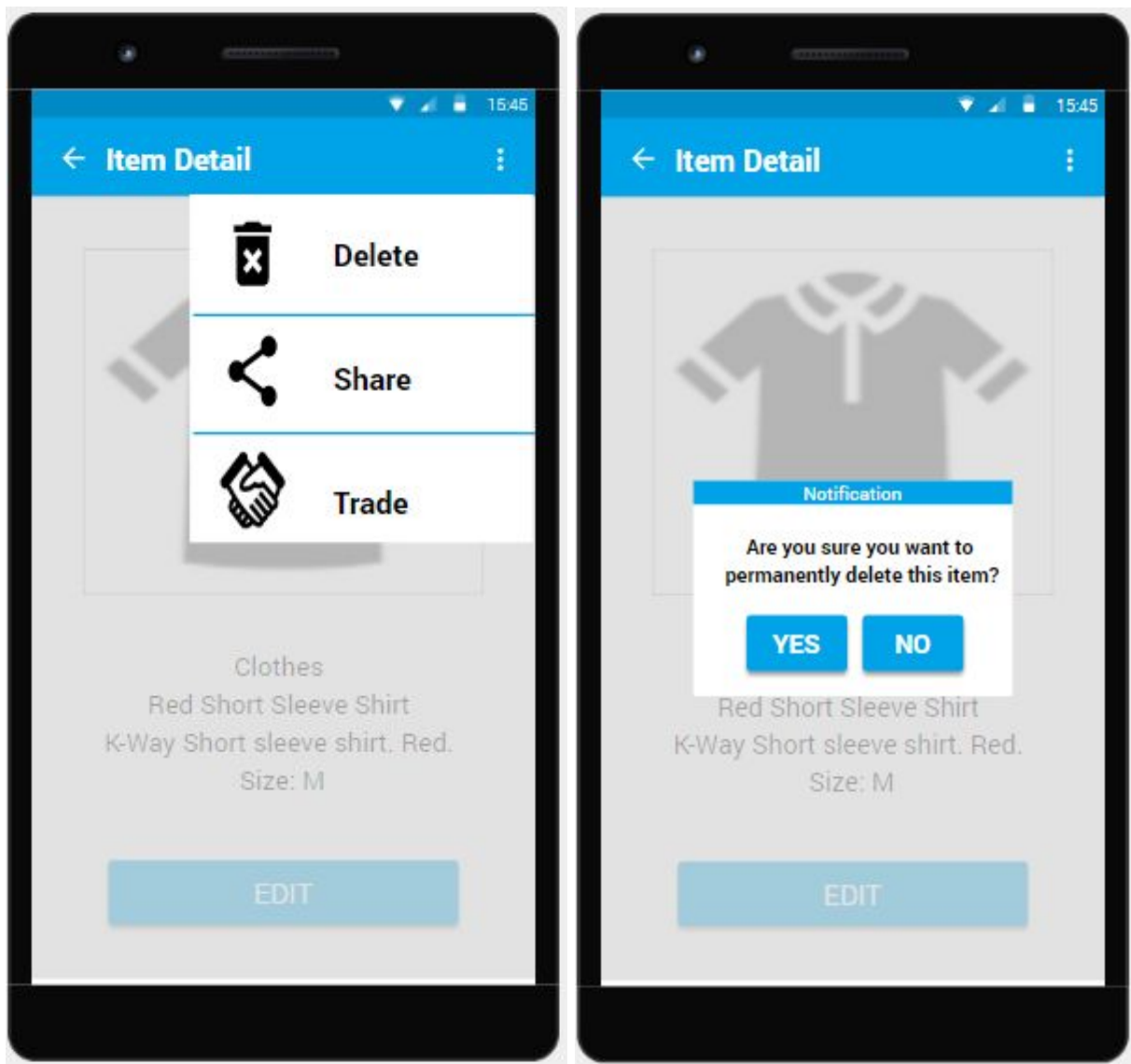
View Item Screen



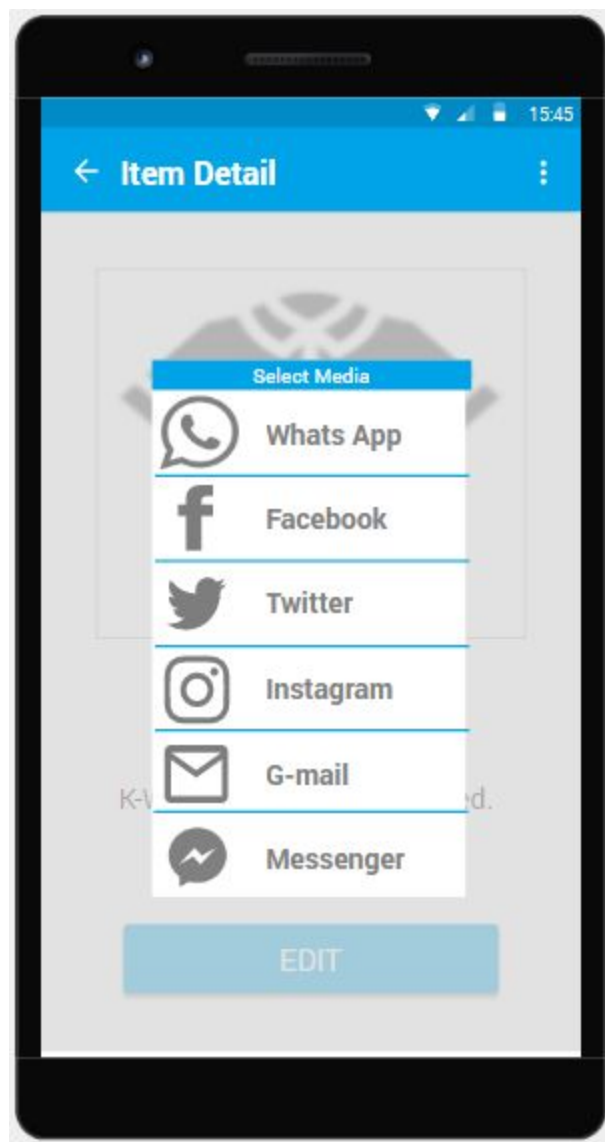
Item Detail Screen



Item Delete Screen



Item Share and Trade Screen



Edit Item Screen



A mobile application screen titled "Edit Item" with a blue header bar. The screen displays a placeholder image of a dark polo shirt. Below the image are three input fields: a dropdown menu for "Category", a text box for "Title", and a larger text box for "Description". A blue "SAVE" button is positioned at the bottom of the form area. The status bar at the top shows the time as 15:45 and various system icons.

← Edit Item

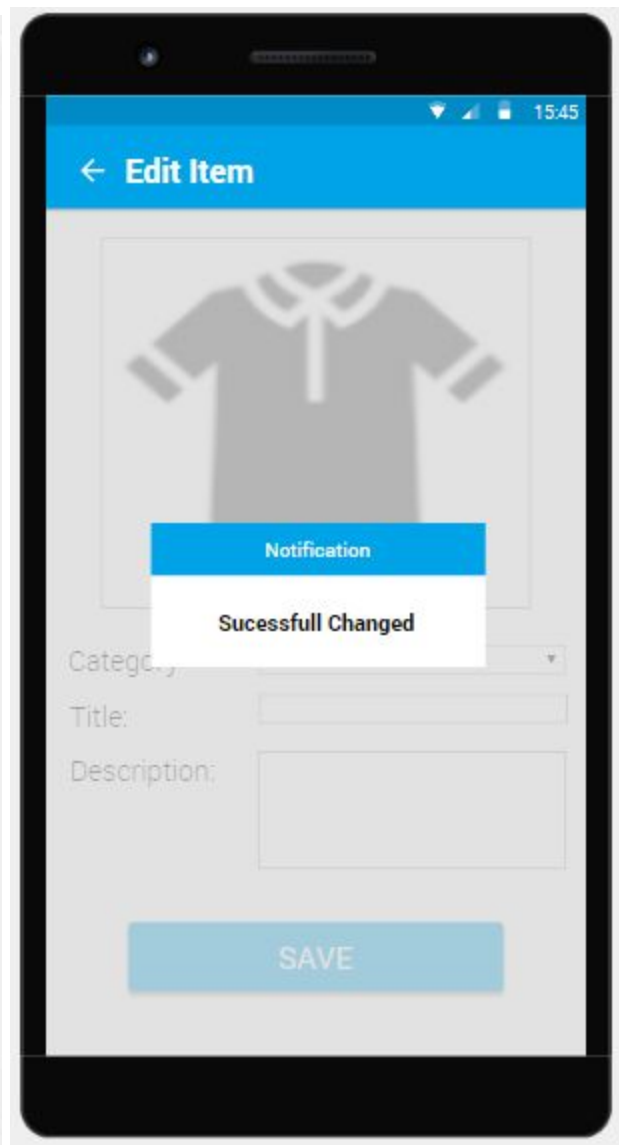


Category

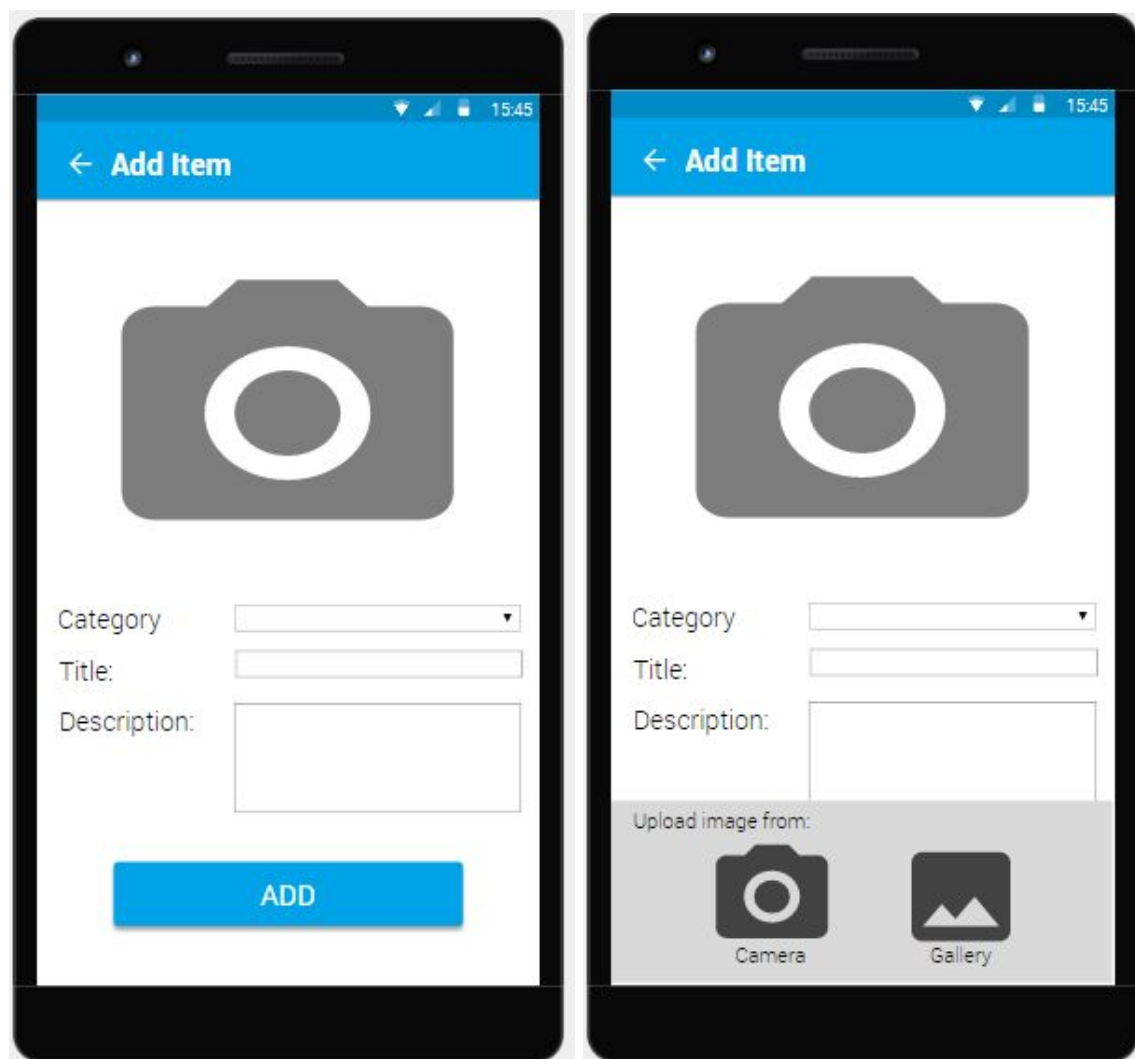
Title:

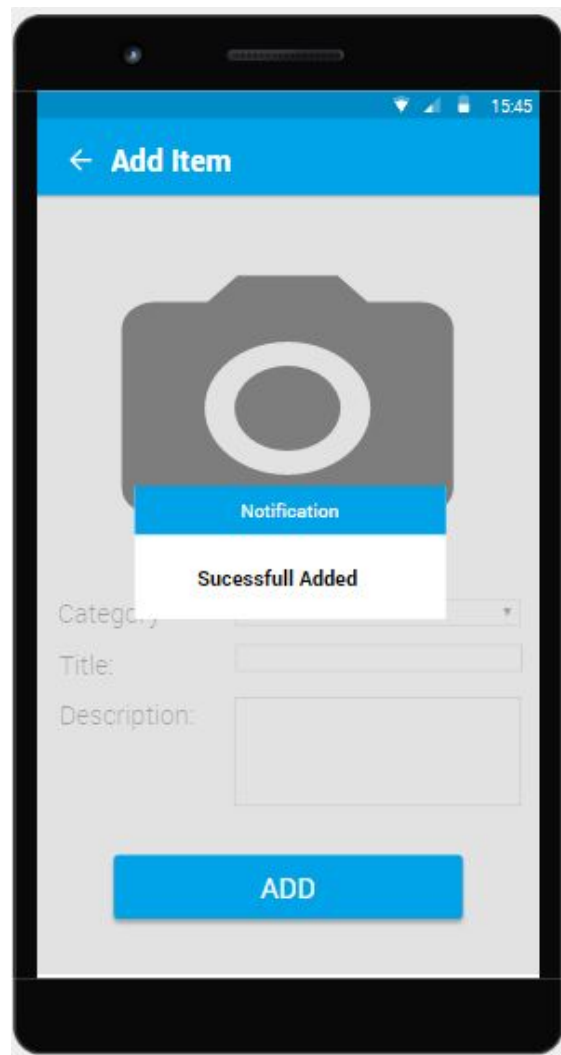
Description:

SAVE

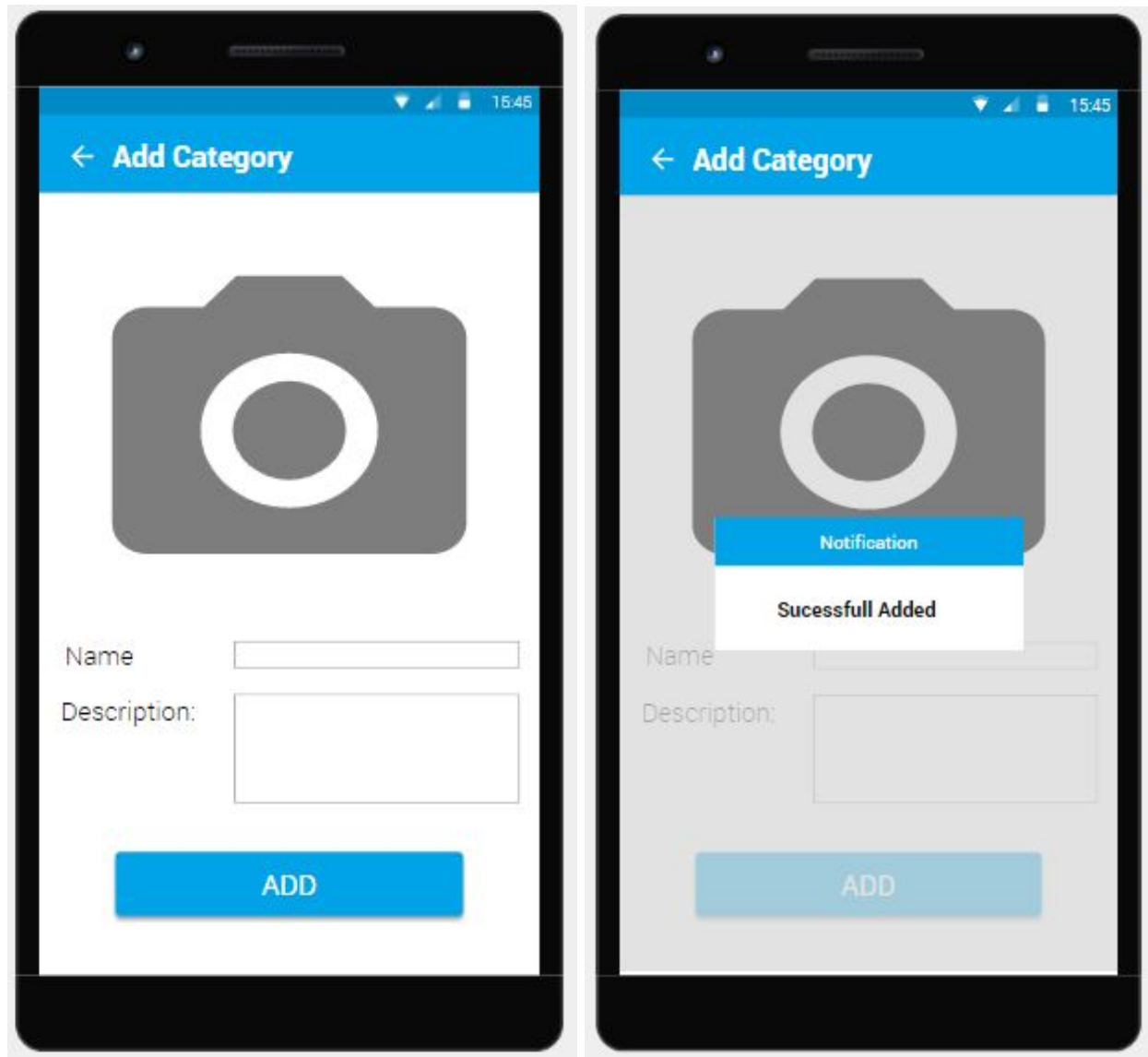


Add Item Screen





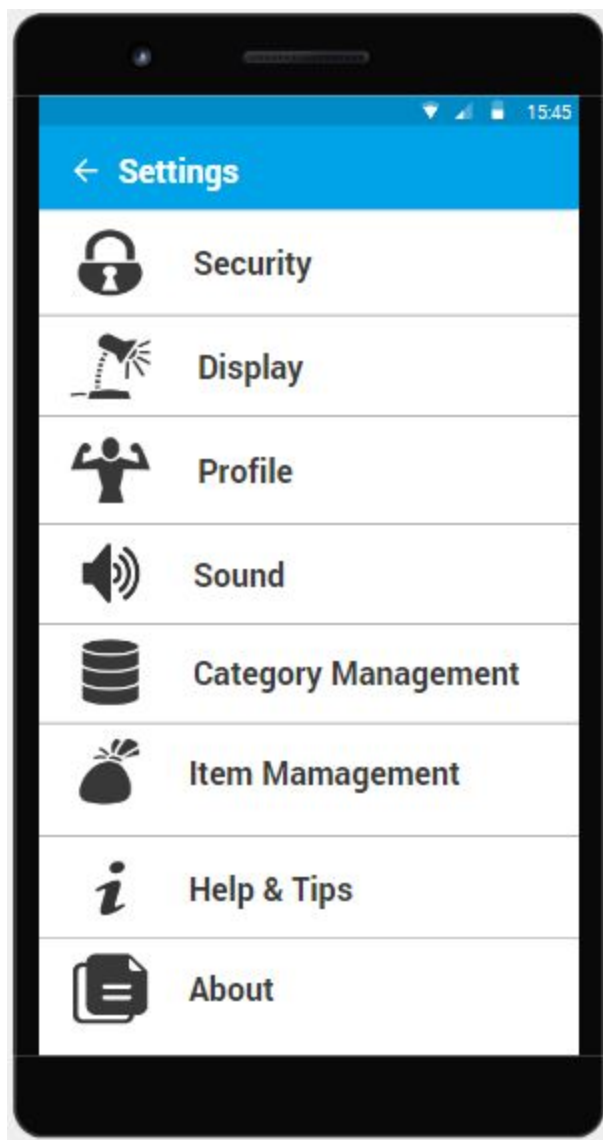
Add New Category Screen



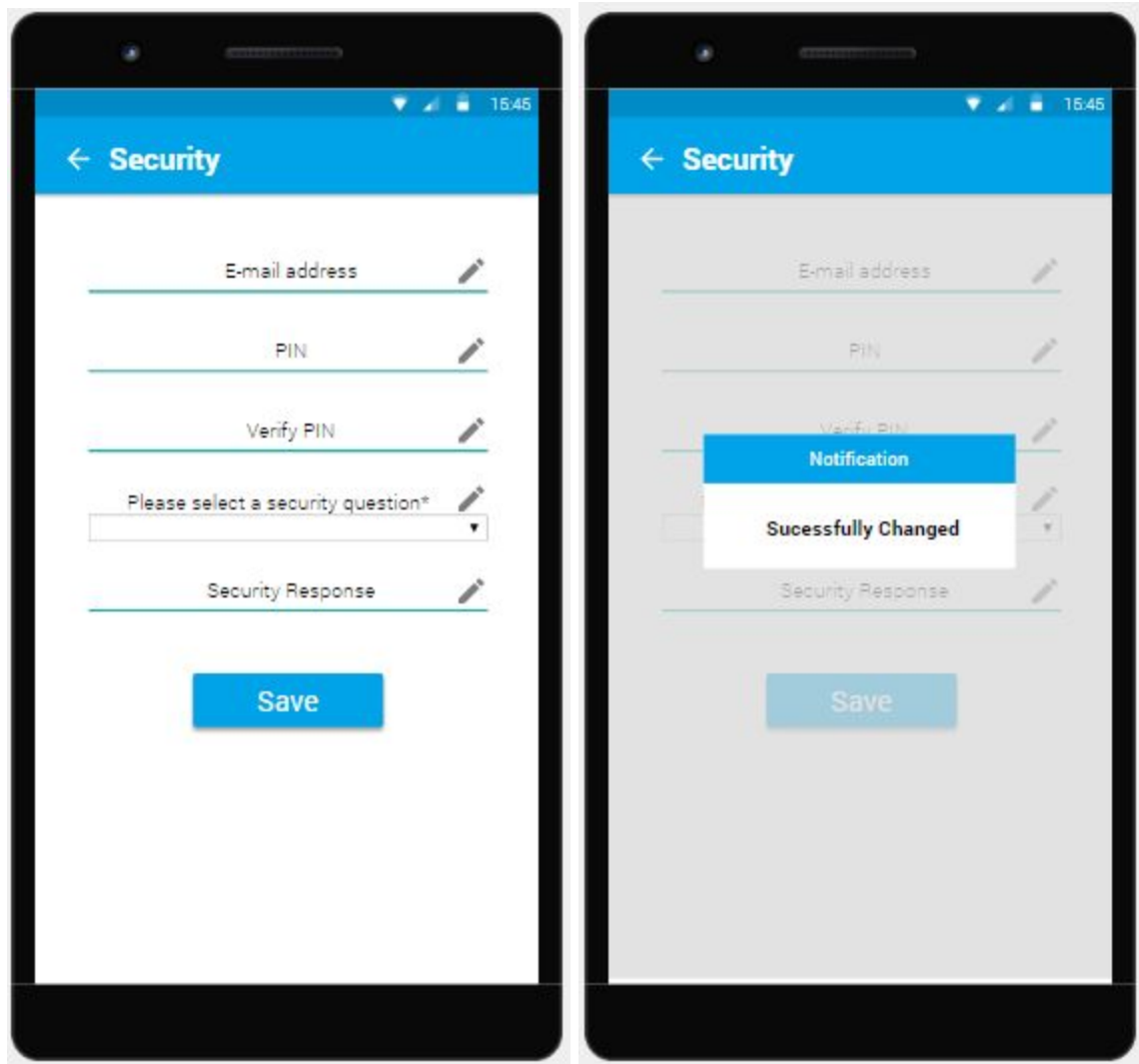
Slideshow Screen



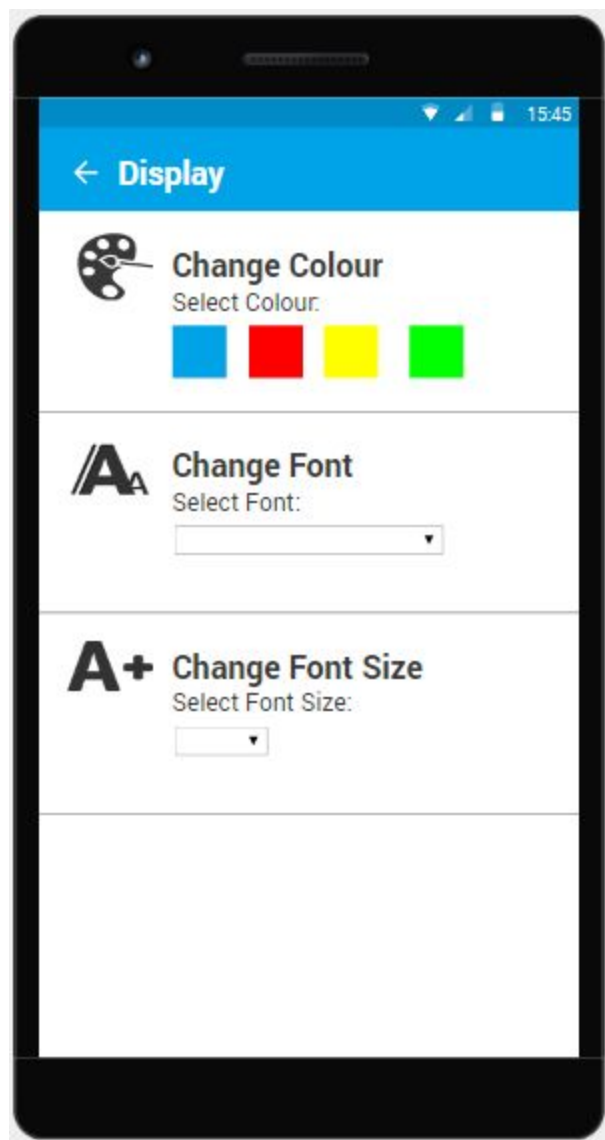
Settings Screen



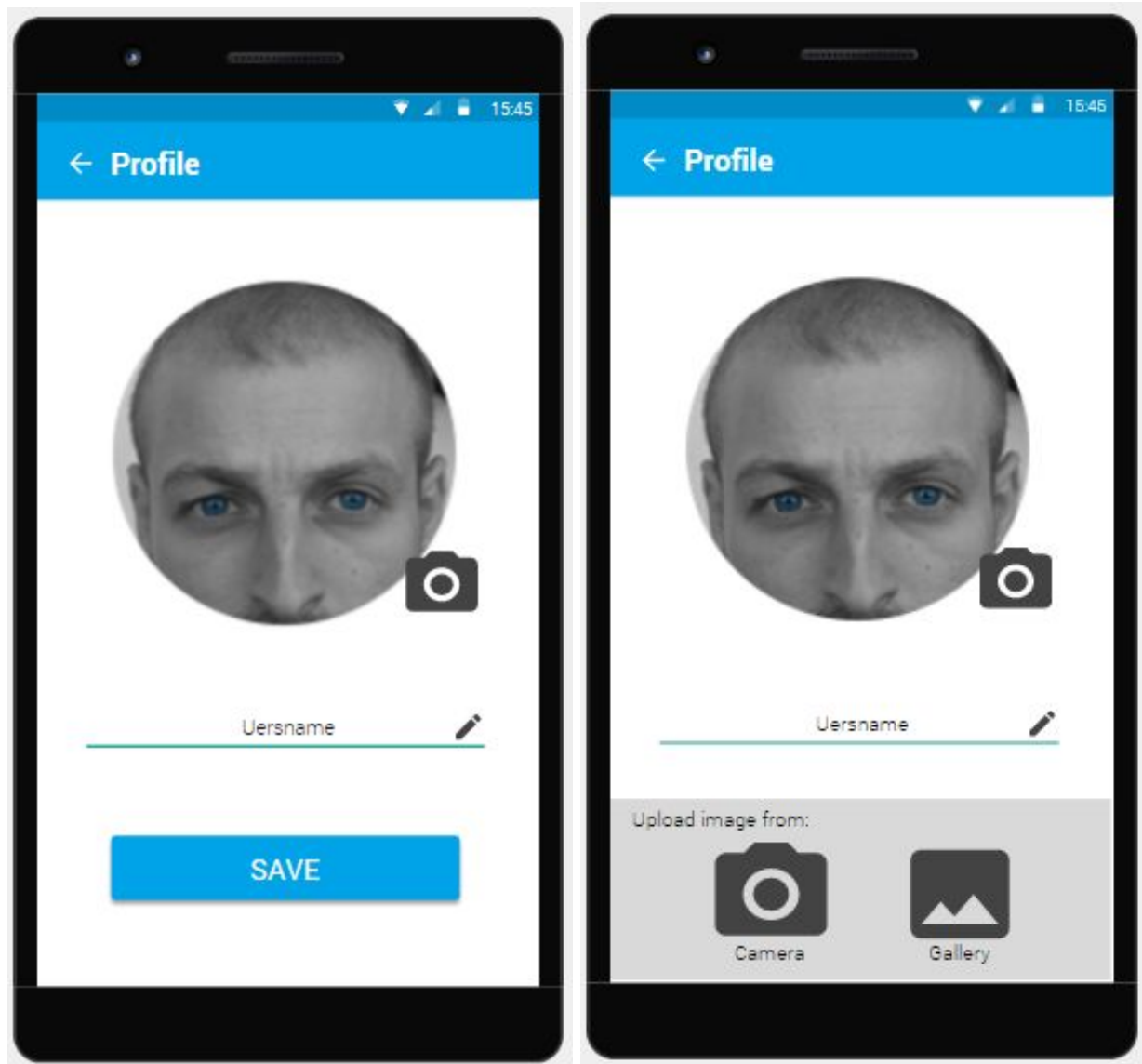
Settings > Security Screen



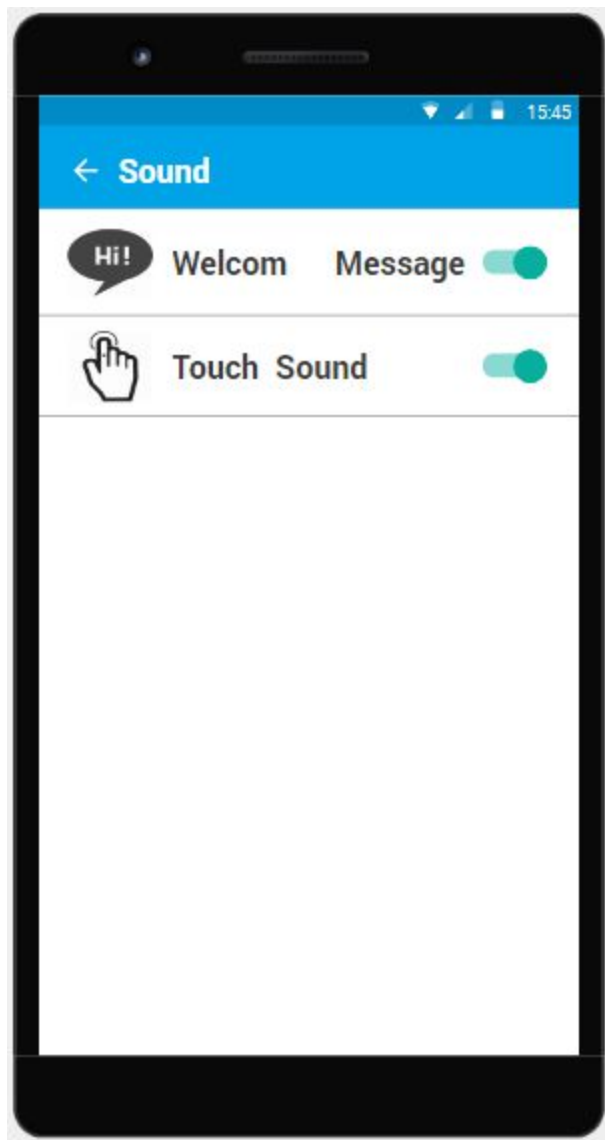
Settings > Display Screen



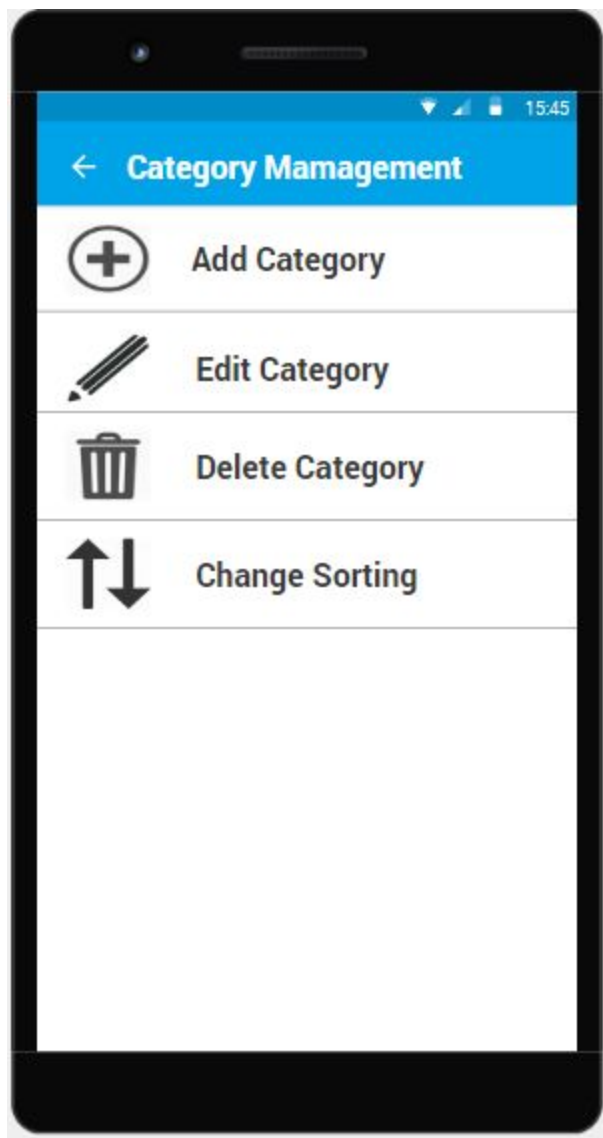
Settings > Profile Screen



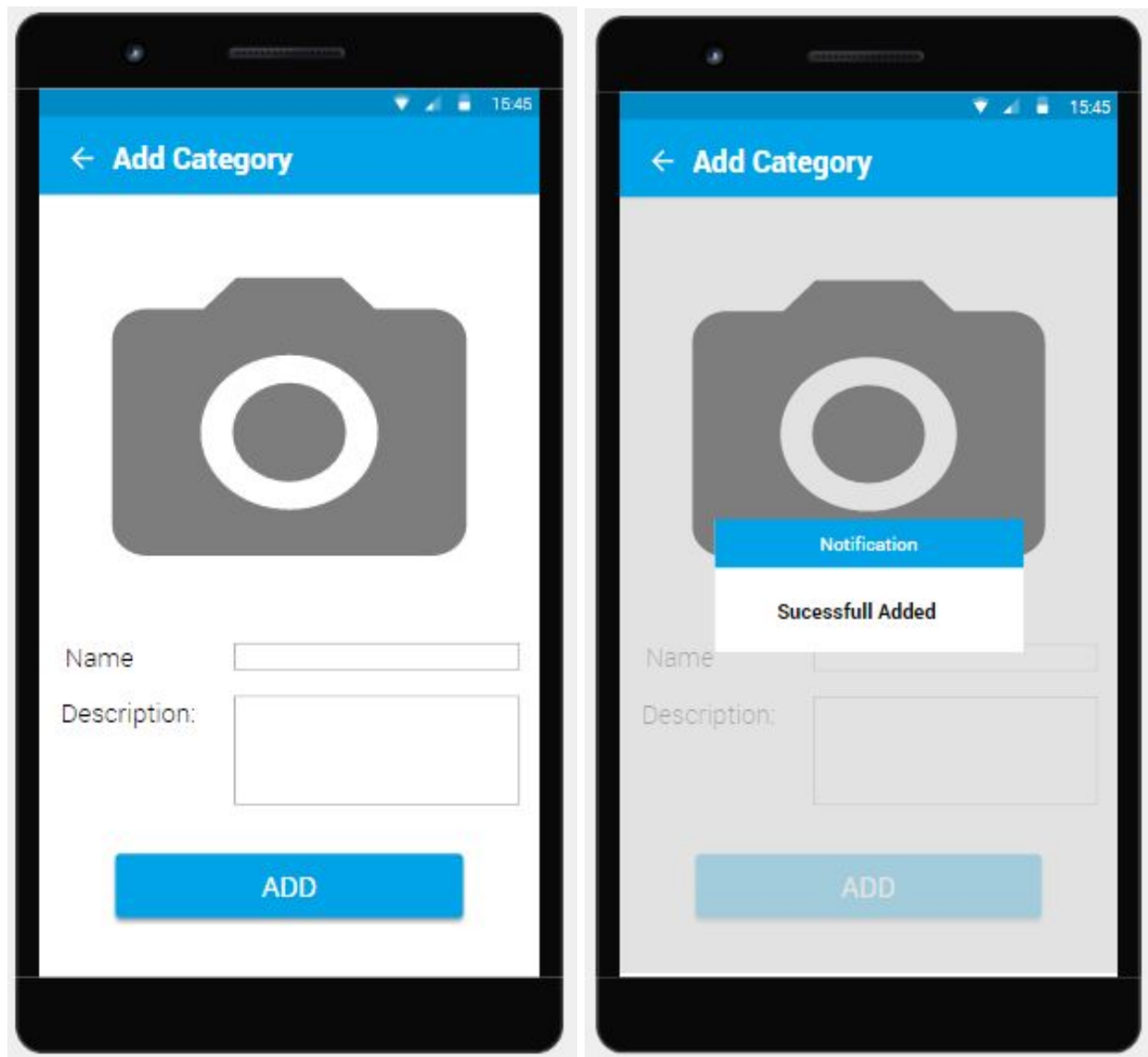
Settings > Sound Screen



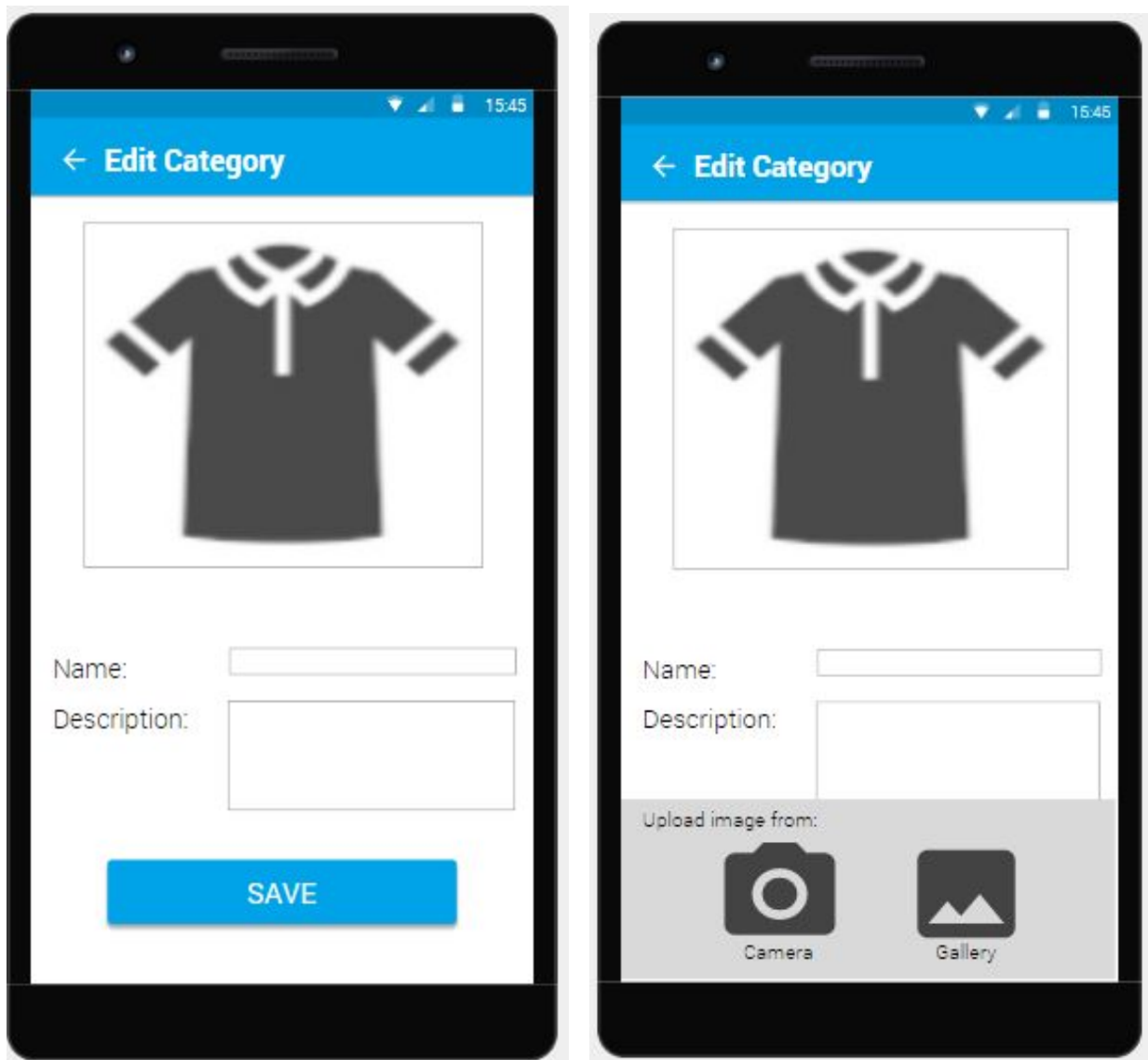
Setting > Category Management Screen

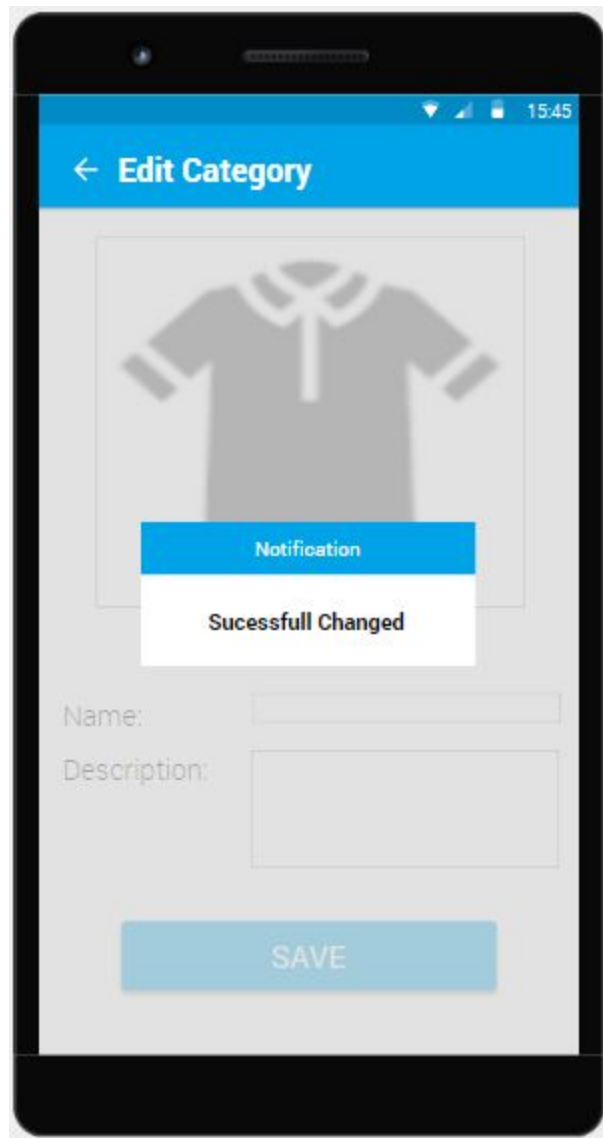


Setting > Category Management > Add Category Screen

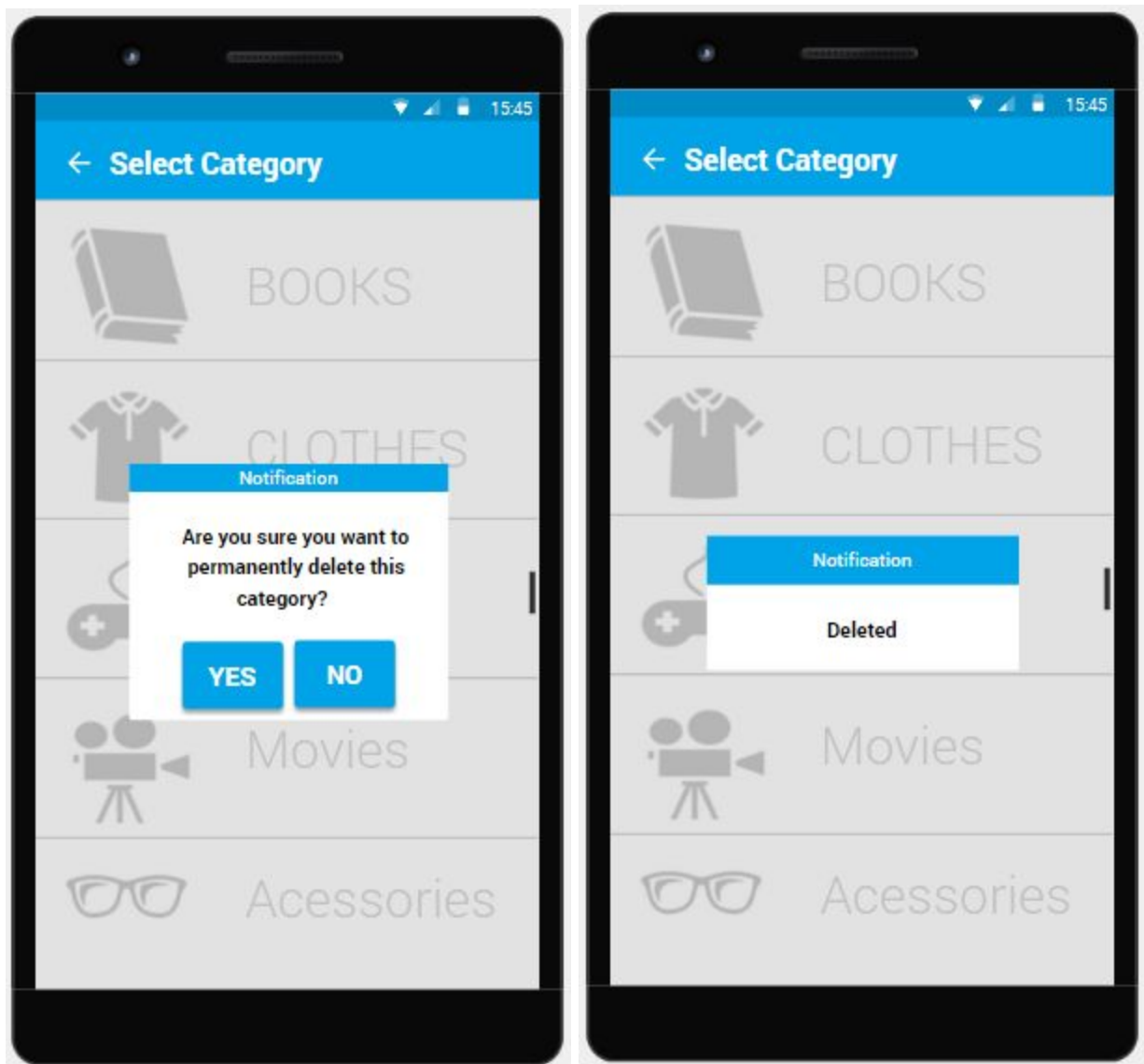


Setting > Category Management > Edit Category Screen

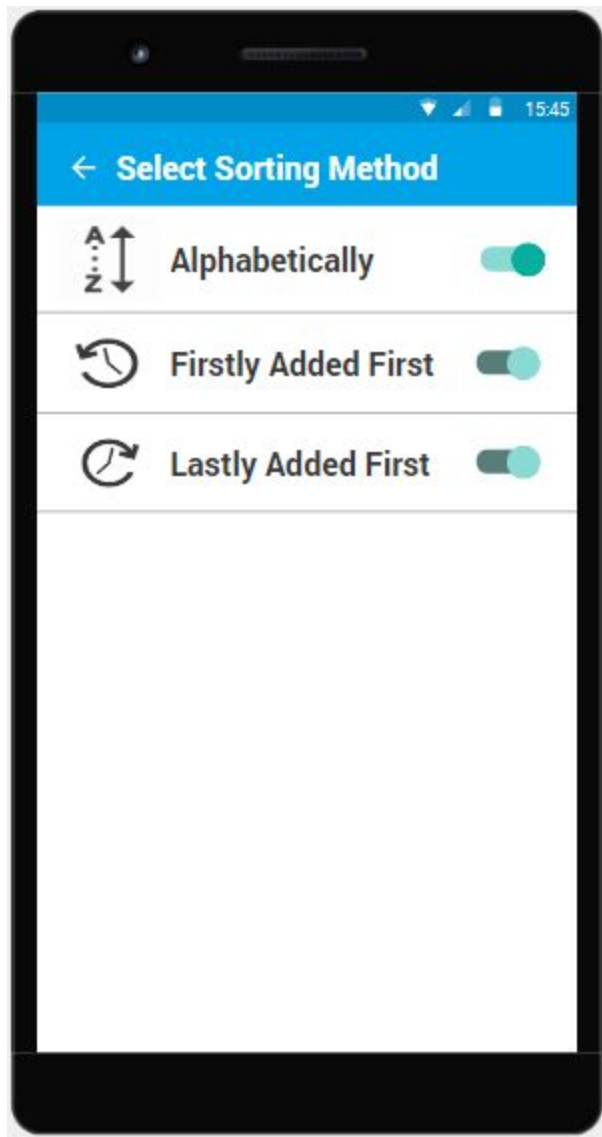




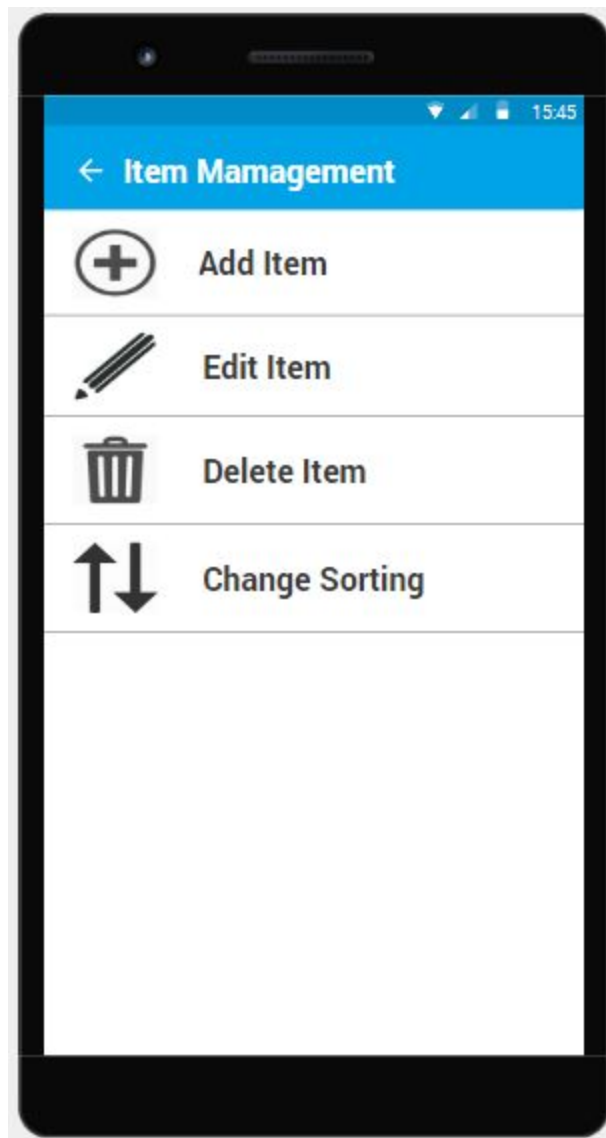
Setting > Category Management > Delete Category Screen



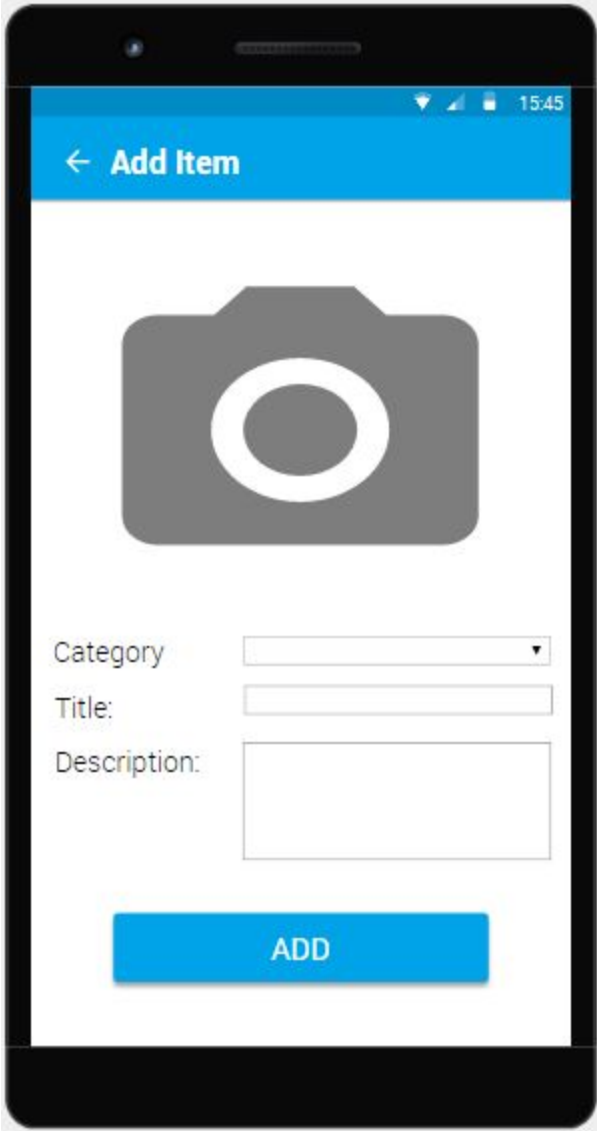
Setting > Category Management > Change Sorting Screen



Setting >Item Management



Setting >Item Management > Add Item Screen



A mobile application screen titled "Add Item" with a blue header bar. The screen features a large gray camera icon for image upload. Below it are three input fields: a dropdown menu for "Category", a text box for "Title:", and a larger text box for "Description:". At the bottom is a blue "ADD" button. The status bar at the top shows the time 15:45 and various icons.

← Add Item

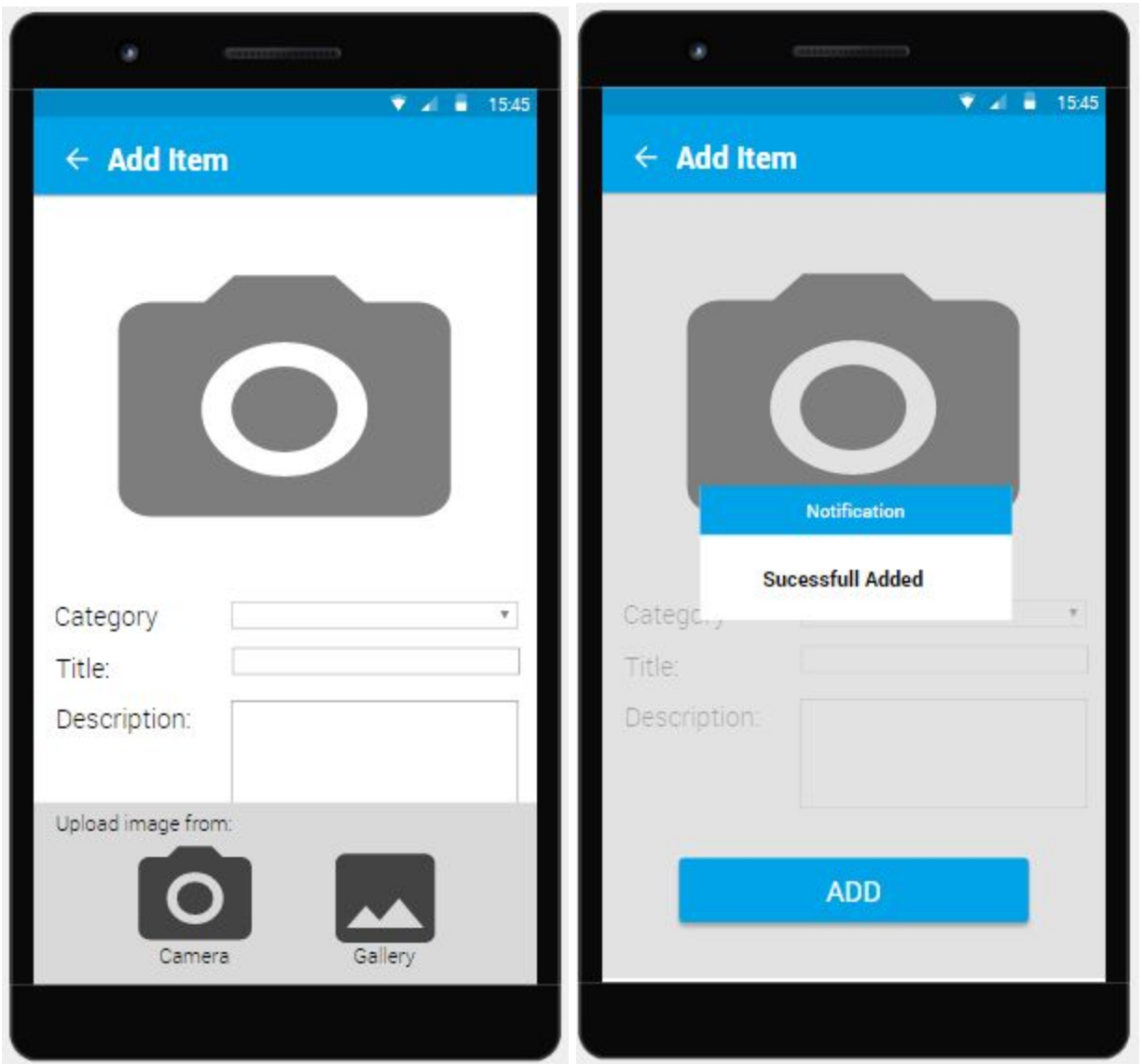
15:45

Category

Title:

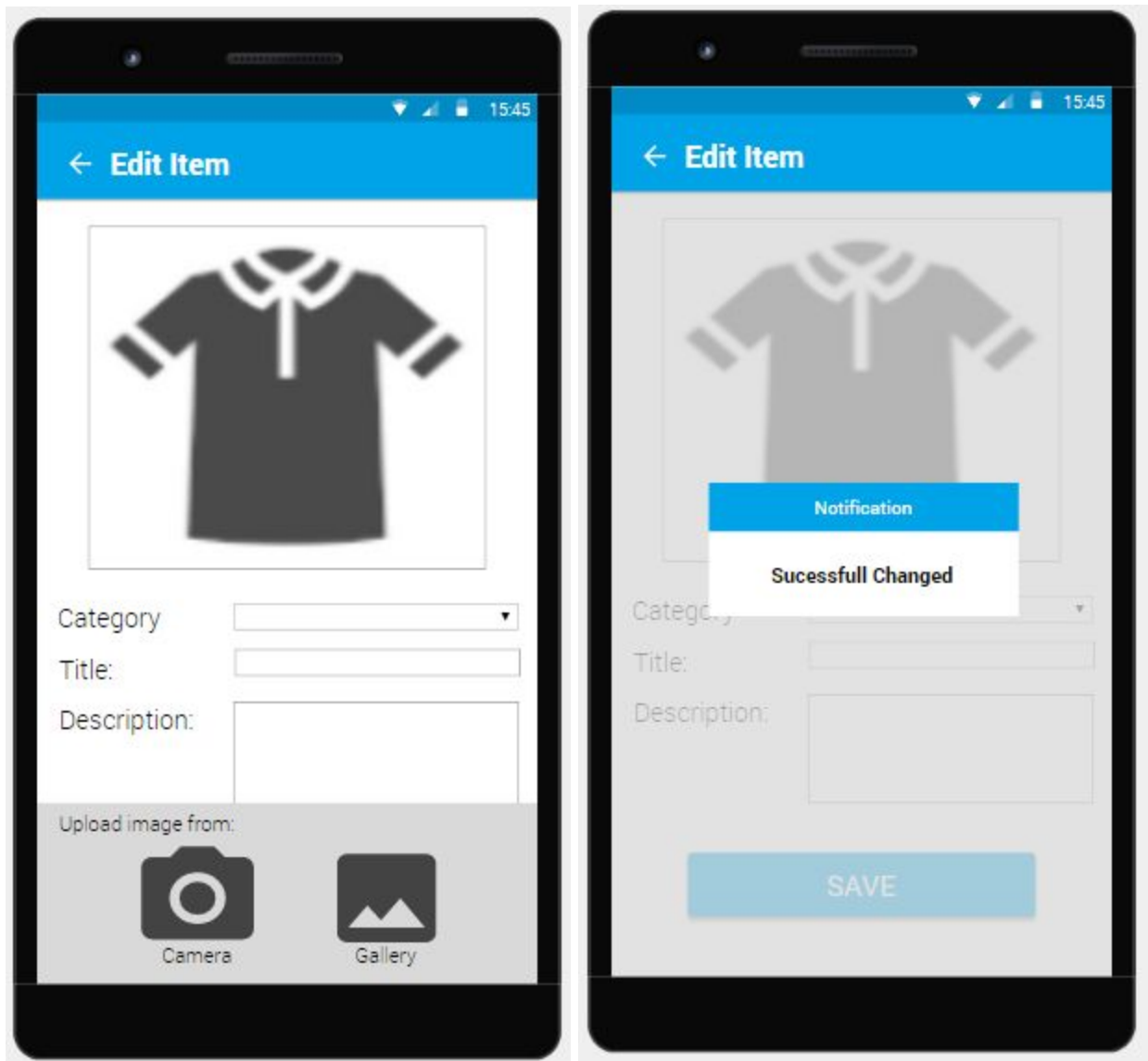
Description:

ADD

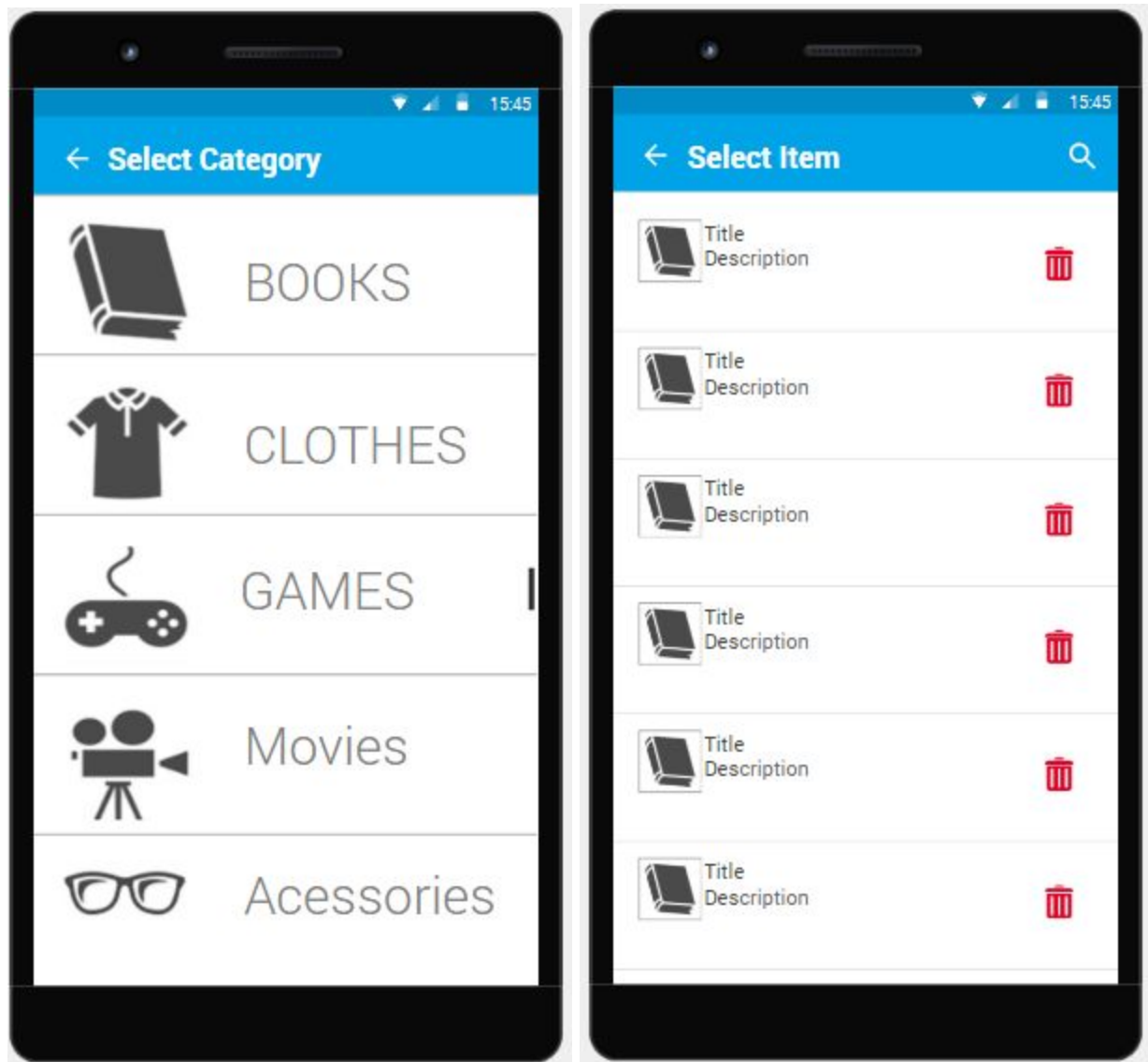


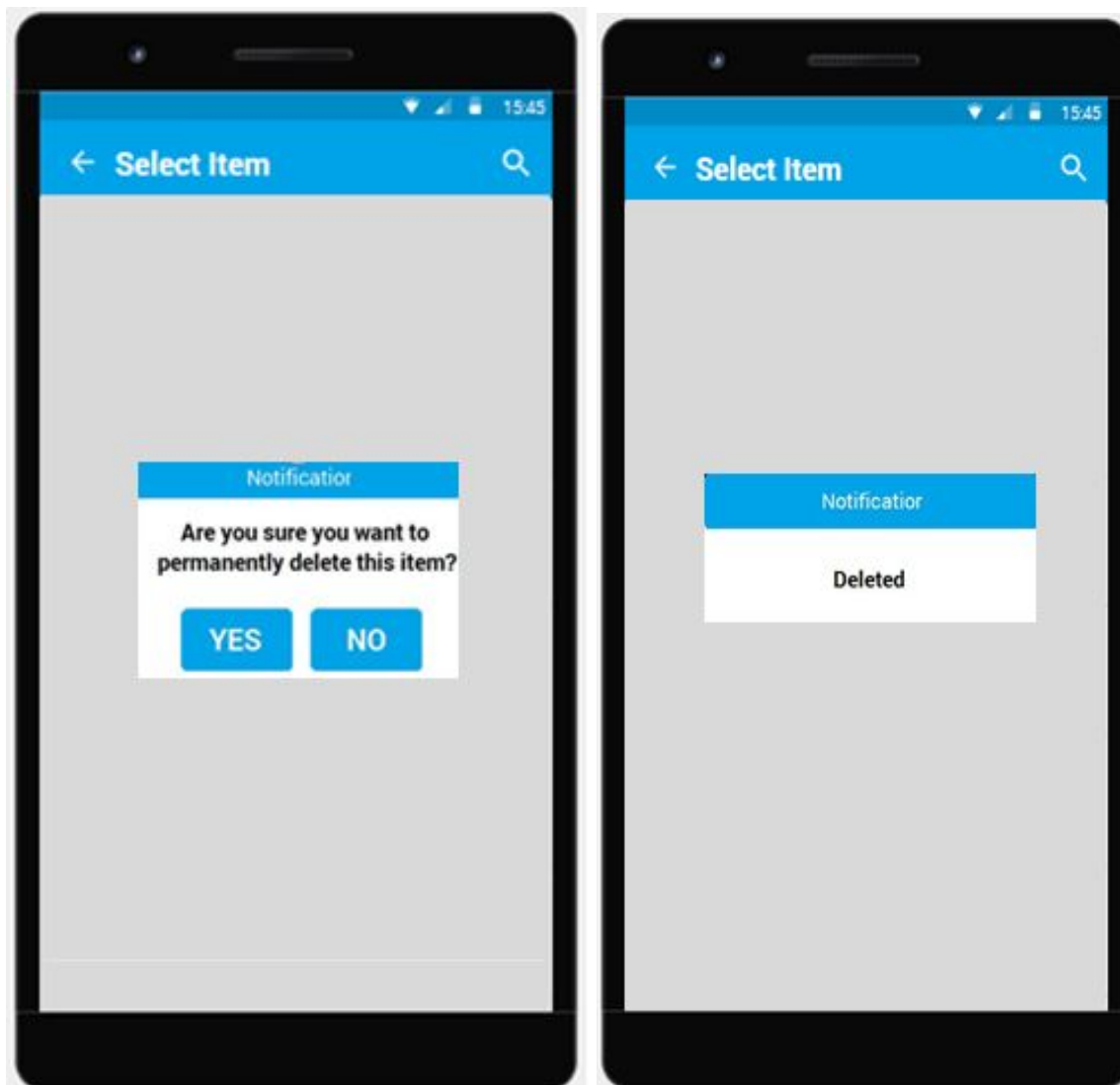
Setting >Item Management > Edit Item Screen



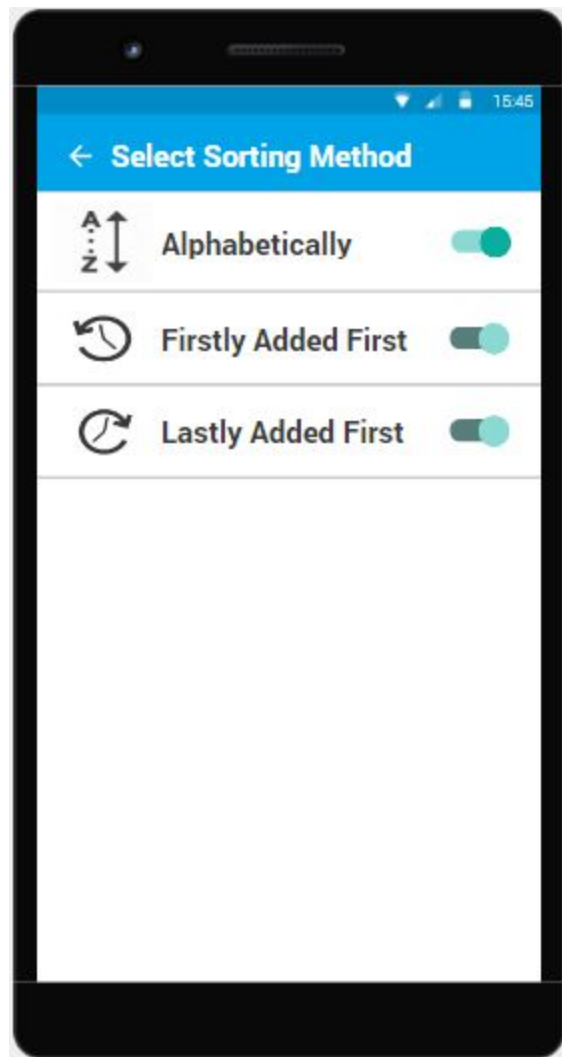


Setting >Item Management > Delete Item Screen

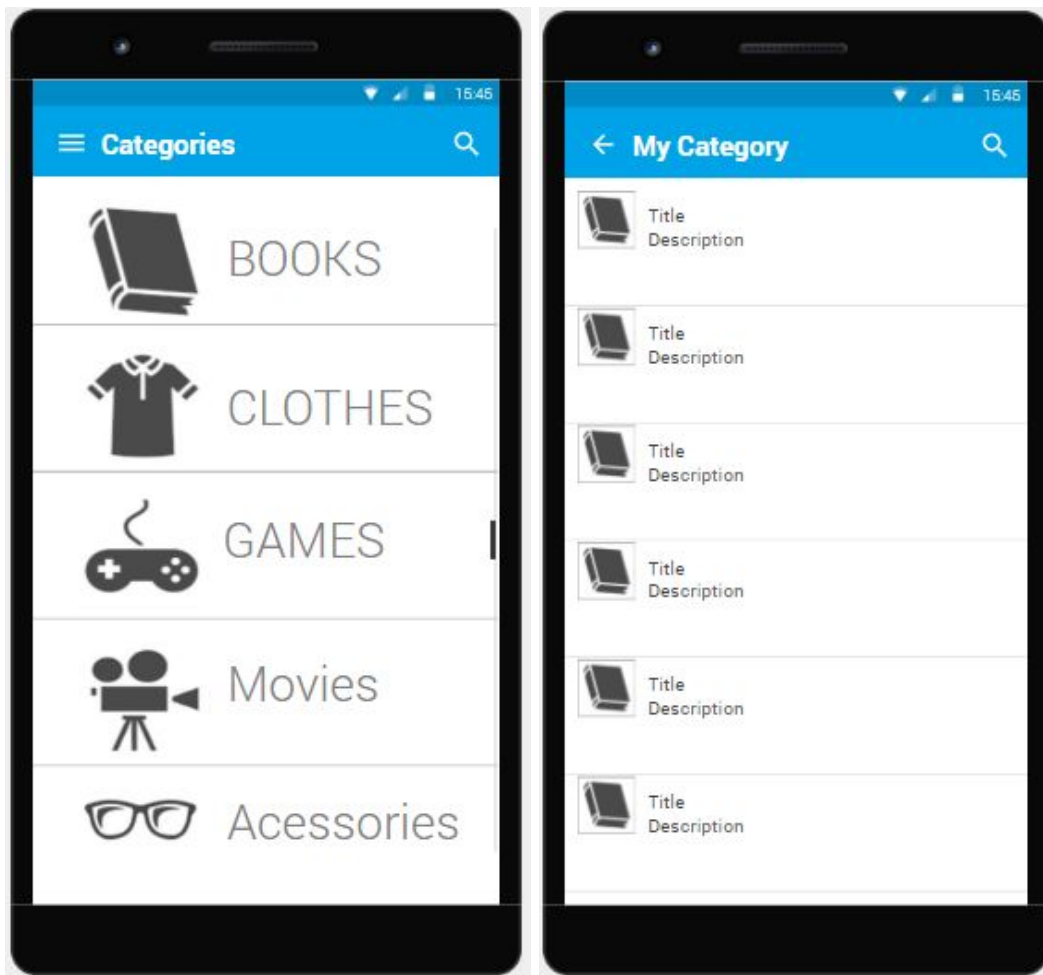


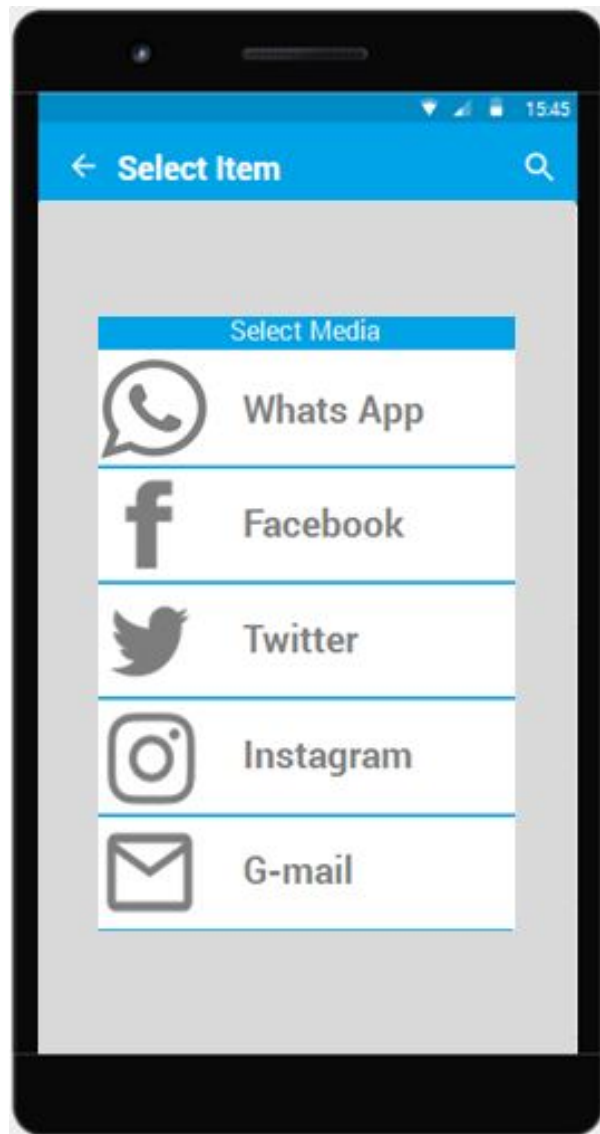


Setting >Item Management > Change Sorting Screen

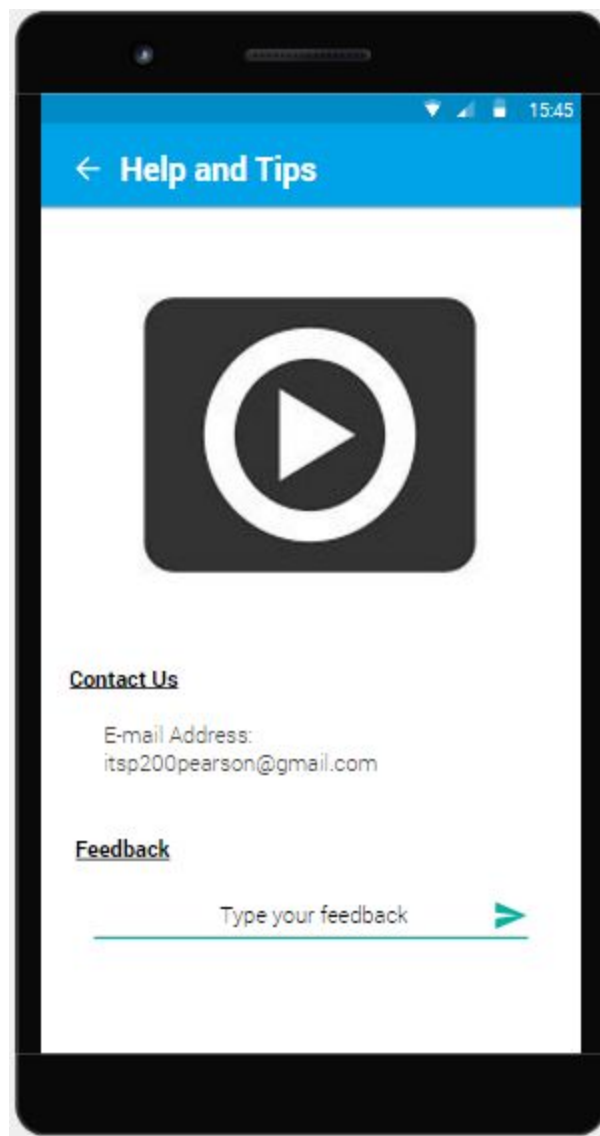


Share and Trade Screen





Help and Tips Screen



About Screen





2. Customer sign-off

Customer name and surname 	Customer signature
Group leader name and surname 	Group leader signature



Appendices

Appendix A

Nielsen's usability heuristics are a set of high-level design principles that are used to evaluate a user-interface to determine whether the interfaces conforms to well established and tested design principles (Preece et al. 2015). A revised version of Nielsen's heuristics are listed below:

- 1) The system should always provide the user with feedback;
- 2) The systems should match the user perception of the real world;
- 3) The system should provide the user with a clear exit to an unwanted state;
- 4) The systems should maintain the same standard and a high a level of consistency throughout;
- 5) The system should seek to eliminate error-prone conditions;
- 6) The system should minimise the load on the user's memory by making options and potential actions more visible;
- 7) The system must be efficient to use;
- 8) The system should make use of a minimalist aesthetic;
- 9) The system should aid users to recognise potential errors and assist the user in accounting for them; and
- 10) The system should provide the user with help documentation (Preece et al. 2015).

These heuristics are intended to be used by designers to compare or evaluate their designs and change their design accordingly. Each iteration of the design should seek to use these guidelines to solve usability problems (Stair & Reynolds 2015).

Bibliography

Android Developers, Create and Manage Virtual Devices | Android Studio. Available at:
<https://developer.android.com/studio/run/managing-avds.html>
[Accessed August 3, 2017].

Android Developers, Download Android Studio and SDK Tools | Android Studio. Available at:
https://developer.android.com/studio/index.html?gclid=CjwKEAju9_jBRCXycSarr3csWcSJABthk071PgsXQmXkW2jaenCzmxk5faXdS3nKbo99zzQizAWmhoC6N_w_wcB
[Accessed June 14, 2017].

IEEE, 1990. *IEEE Standard Glossary of Software Engineering Terminology*, Inst of Engineering, Electrical & Electronic Standards.

Oracle, Java SE Development Kit 8 - Downloads. Available at:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
[Accessed August 3, 2017].

Oracle, Java SE Runtime Environment 8 - Downloads. Available at:
<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>
[Accessed August 3, 2017].

Dennis, A., Wixom, B. H. & Tegarden, D., 2005. *Systems Analysis and Design w/ltH UML Version 2.0*. 2nd ed. New Jersey: John Wiley & Sons, Inc.

Satzinger, J. W., Jackson, R. B. & Burd, S. D., 2015. *Systems Analysis and Design in a Changing World*. 7th (E-BOOK) ed. Massachusetts: Cengage Learning.


Shelly, G. B., Cashman, T. J. & Rosenblatt, H. J., 2003. *Systems Analysis and Design*. 5th ed. Boston: Thomson Course Technology.

Bertolino, A., 2001. Chapter 5: Software Testing. *University of Houston-Clear Lake*. Available at:
http://sce.uhcl.edu/helm/SWEBOK_IEEE/data/swebok_chapter_05.pdf
[Accessed August 7, 2017].

Connolly, T.M. & Begg, C.E., 2005. *Database Systems: A Practical Approach to Design, Implementation, and Management*, Pearson Education.

Finley, K., 2012. What Exactly Is GitHub Anyway? *TechCrunch*. Available at:
<http://social.techcrunch.com/2012/07/14/what-exactly-is-github-anyway/>
[Accessed June 14, 2017].

Justinmind, Prototyping tool for web and mobile apps - Justinmind. Available at:
<https://www.justinmind.com>
[Accessed June 14, 2017].

- 
- Mahfuj, M.M., 2012. Logical design vs physical design. Available at: <https://www.slideshare.net/MahediMahfujAnik/logical-design-vs-physical-design> [Accessed May 17, 2017].
- Preece, J., Rogers, Y. & Sharp, H., 2015. *Interaction Design: Beyond Human-Computer Interaction*, John Wiley & Sons.
- softwaretestingclass, 2012. System Testing: What? Why? & How? *Software Testing Class*. Available at: <http://www.softwaretestingclass.com/system-testing-what-why-how/> [Accessed June 15, 2017].
- Stair, R. & Reynolds, G., 2015. *Principles of Information Systems*, Cengage Learning.
- Williams, L., 2006. Testing Overview and Black-Box Testing Techniques. *North Carolina State University*. Available at: <http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf> [Accessed August 7, 2017].
- WebAIM, 2016. Microsoft Word: Creating Accessible Documents <http://webaim.org/techniques/word/> [Accessed August 9, 2017].