Software Requirements Specification

for

Pro Bikes

Version 1.2

Prepared by

Group Name: Group 3

|  |  |  |
| --- | --- | --- |
| Jaime Jahuey | (864) 381-2982 | jahuey@email.uscupstate.edu |
| David Krechko | (864) 381-2158 | krechkod@email.uscupstate.edu |
| Daniel Espina | (864) 380-4588 | espina@email.uscupstate.edu |
| Alfonso Quistian | (864) 384-6231 | quistian@email.uscupstate.edu |

|  |  |
| --- | --- |
| Instructor: | Dr. Schwartz |
| Course: | CSCI540 |
| Date: | December 4, 2015 |

Contents

Revisions ii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3 Specific Requirements 5

3.1 External Interface Requirements 5

3.2 Functional Requirements 6

3.3 Behaviour Requirements 6

4 Other Non-functional Requirements 6

4.1 Performance Requirements 6

4.2 Safety and Security Requirements 7

4.3 Software Quality Attributes 7

5 Other Requirements 7

Appendix A – Data Dictionary 8

Appendix B - Group Log 9

# 

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | Jaime Jahuey  David Krechko  Daniel Espina  Alfonso Quistian | Primary Draft | 10/02/15 |
| 1.1 | Jaime Jahuey  David Krechko  Daniel Espina  Alfonso Quistian | Second Draft | 10/26/15 |
| 1.2 | Jaime Jahuey  David Krechko  Daniel Espina  Alfonso Quistian | Third Draft | 11/11/15 |

# Introduction

*This section will describe a general view of the software Pro Bikes.*

## Document Purpose

The purpose of this document is to describe the functionality of an android mobile application for the management of a bicycle shop. This software will have the functionalities to manage inventory, repairs, and profits. The bicycle shop owner will able to manage his shop from his mobile device. The main purpose of this mobile software application is to make the management feasible for the owner.

## Product Scope

The application Pro Bikes will only require an android phone or device. The application will create a local database on the user’s device and will store all of his information in the device. The installation of this application on his device will make it more feasible for him to manage his bike shop on the go since the application at his convenience on his mobile phone or device.

## Intended Audience and Document Overview

This documentation is mainly intended for the bike shop owner or any other bike shop owner that would like to use this application. This application will only be used by on person and it’s not intended for customers.

## Definitions, Acronyms and Abbreviations

Bit values- Values that consist of 1s and 0s where 1 mean “yes” and 0 means “no” in this project.

Database- a structured set of data held in a computer, especially one that is accessible in various ways.

Tables- a set of data elements using a model of vertical columns and horizontal rows. This is used to store information for the database.

Dialog boxes- a small area on screen, in which the user is prompted to provide information or select commands.

Radio button- an icon representing one of a set of options, only one of which can be selected at any time.

Textboxes-An on-screen rectangular frame into which you type text.

## Document Conventions

This document follows the IEEE formatting requirements.

## References and Acknowledgments

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.

TO DO: Use the standard IEEE citation guide for this section. An example citation guide is posted for you on the website.>

# Overall Description

## Product Perspective

This software a new, self-contained product that is design to be used only by any bicycle store owner to easily manage his store. The owner will be able to manage his inventory such as adding and removing bicycles form the database. He will also add sold bikes to a table and calculate profit. Repairs and task will be managed for the store owner so he would never miss a due date. The application will provide a beneficial experience for the owner and his consumers.

## Product Functionality

The android application will let the bike shop owner manage his bike shop in a very easy and efficient way. The application will help the bike owner in the following ways.

* Management
  + The user will be able to add, remove and display his/her inventory.
* Repairs
  + The user will be able to add, remove, delete and display active and completed repairs.
* Sales
  + The user will be able to sell bikes and view profits.

## Users and Characteristics

The main user of this software will be the bike shop owner. The owner will have extended experience in the bicycle industry and is assumed to know how to use the application. He/she may be the only person who will use the product.

## Operating Environment

This application will be an android application, which can only be used on any android system. Internet connection will not be required expect for downloading the application. The android device need a minimum of SDK Version 16 to run the application but It is recommended to have SDK Version 21.

## Design and Implementation Constraints

The application cannot be installed on any device expect for android. The phone must be in service in order to be able to the call the customer. In order to use the email function, the device must have access to the internet. There is limited memory due to the database being stored on the phone. The android device must need at least SDK Version 16.

## User Documentation

This application is intended to be easy to use. It’ll have user friendly functions like image icons for the buttons and messages that will display if the user inputted information incorrectly. The user manual will have sections that will describe what each button does and what information is required. It’ll also describe how to manage the information stored in the application.

## Assumptions and Dependencies

Major assumptions

* The bike shop owner will be the only one using this application
* Other people such as friends or children will not have access to this device so they cannot accidently add or sell bikes for instance.
* The owner will have an android device with the appropriate requirements in order to use this application.

# Specific Requirements

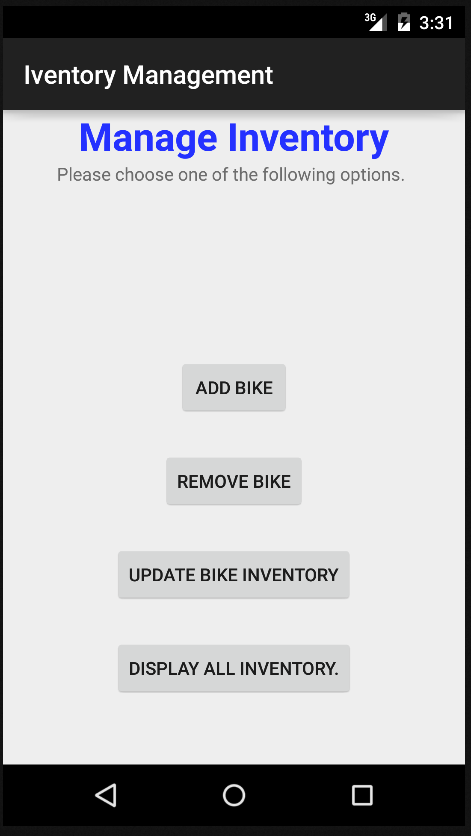
## External Interface Requirements

### User Interfaces

### C:\Users\David\Downloads\PhoneScreens\HomeScreen.PNG

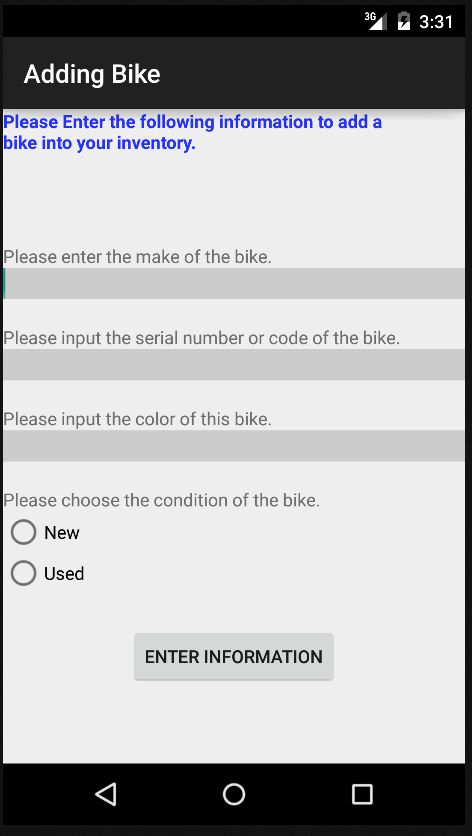
**Figure 1.** Whenever the user opens the application, assuming the app was closed, the screen pictured above will appear with three buttons:

* + Manage Inventory Button
    - * This button will take the user to another screen called “Inventory Management”.
  + Sales Button
    - This button directs the user to another screen called “Sales”.
  + Repairs/Tasks Button
    - This button will take the user to a screen called “Repairs/Tasks”.



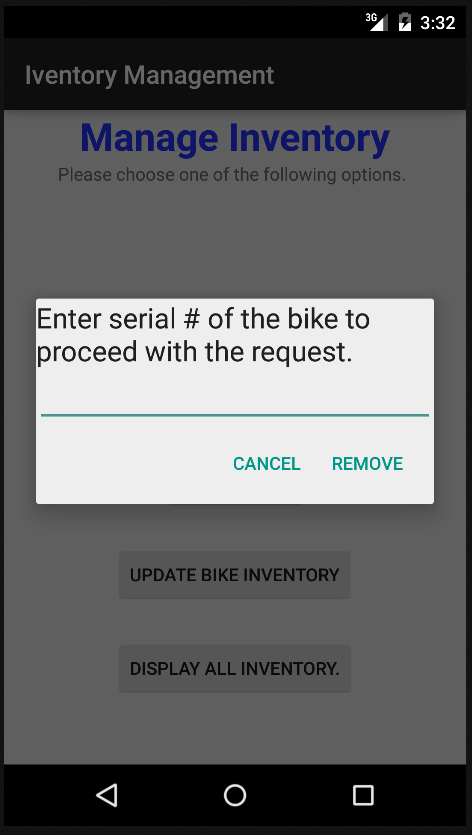
**Figure 2.** The manage inventory section will contain the main Manage Inventory page with all the buttons that leads to pages or dialog boxes. Each button has a specific task that is essential for organizing the bike owner’s inventory.

* Add Bike Button
  + Adds bicycles to the inventory table
* Remove Bike Button
  + Removes bicycles from the inventory table
* Update Bike Inventory
  + Allows the user to edit the inventory table
* Display All Inventory Button
  + Displays the inventory table

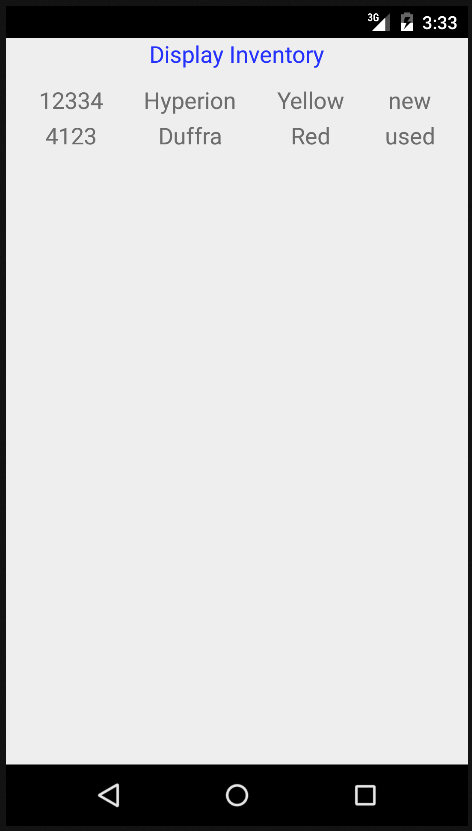


***Figure 3.*** When the Add Bike button is pressed a new screen will pop up on his device that will display text boxes to insert information on the bike he wants to add to the current inventory and a radio button for the condition of the bike (new or used). He will be asked for the following information:

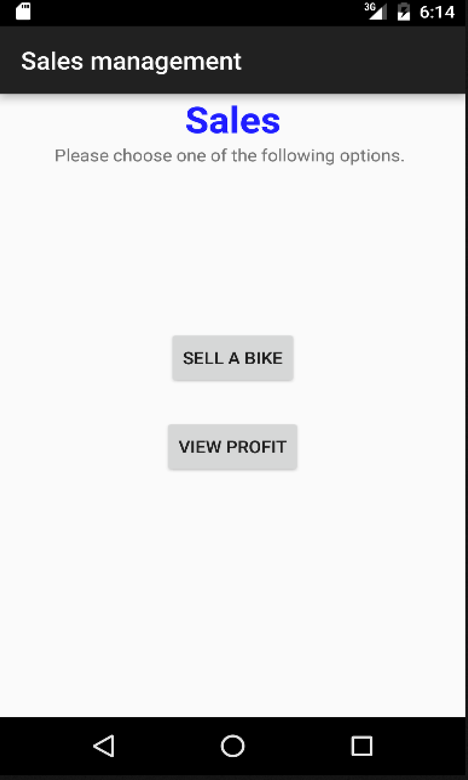
* + - * + The make of the bike
        + The serial number
        + The color



**Figure 4.** The remove button will display a dialog box asking for the bike’s serial number that you want to remove. This will change the status (or bit value) of the item in the table from 1 to 0.

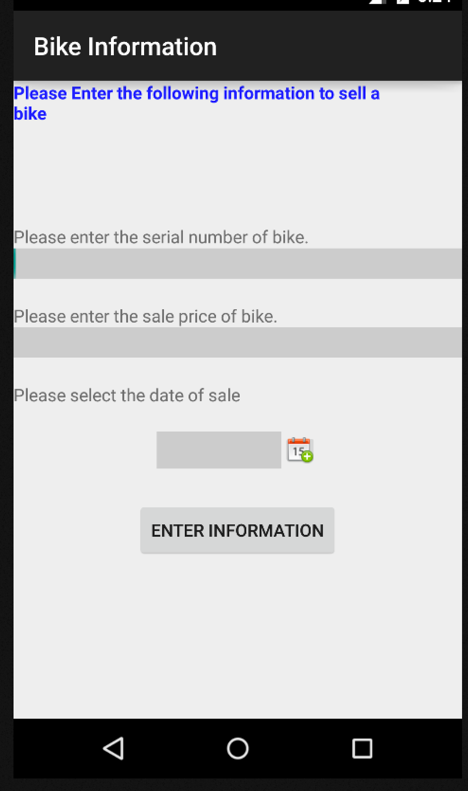
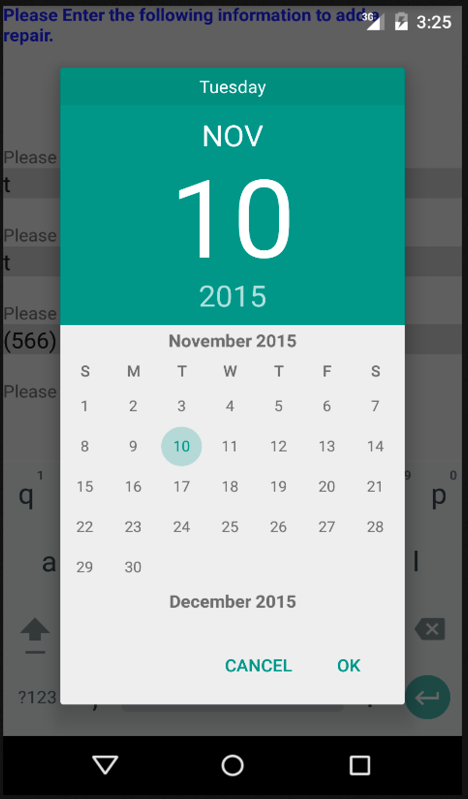


***Figure 5.*** The “Display all inventory” button will allow the user to view all the bikes currently in the inventory. All the information from the Inventory table will be displayed in a list on a new page.



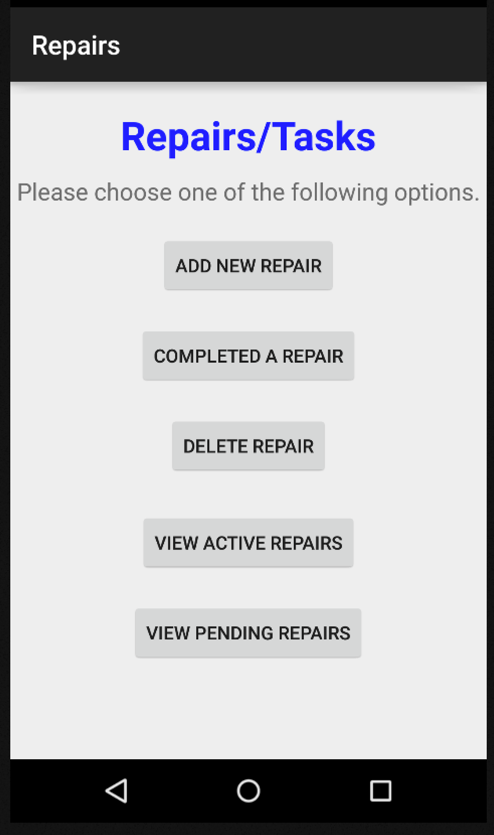
**Figure 6.** This page will pop up once the button “Sales” Button is pressed. This page will have two buttons:

* + Sell A Bike Button
    - * This button will open a new page where the user can record selling a bicycle.
  + View Profit Button
    - * This button will open a new page that will display all the profits made.

****

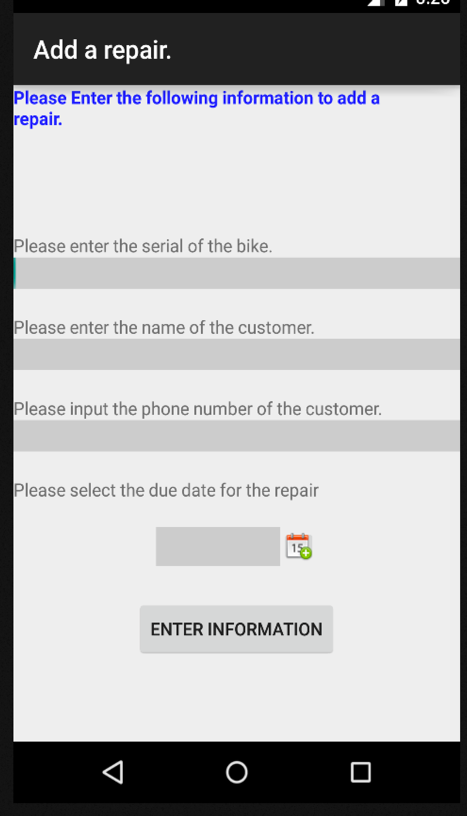
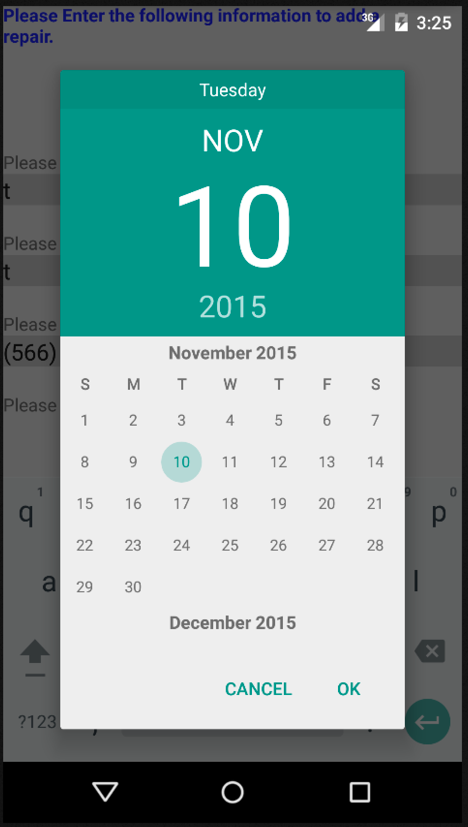
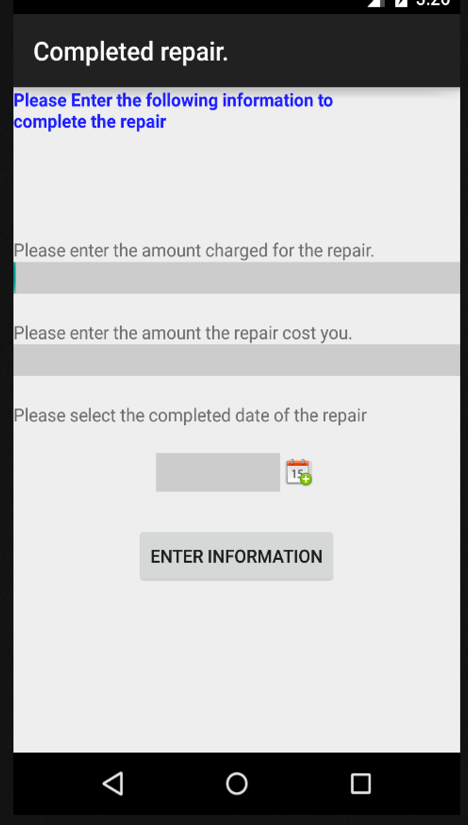
***Figure 7.*** This button will take user to a new page and prompt user to enter information about the bicycle sold. In order to sell, the bicycle must exist in inventory. He will be asked the following information:

* Serial number of the bicycle
* The sale price of the bicycle
* The date of sale



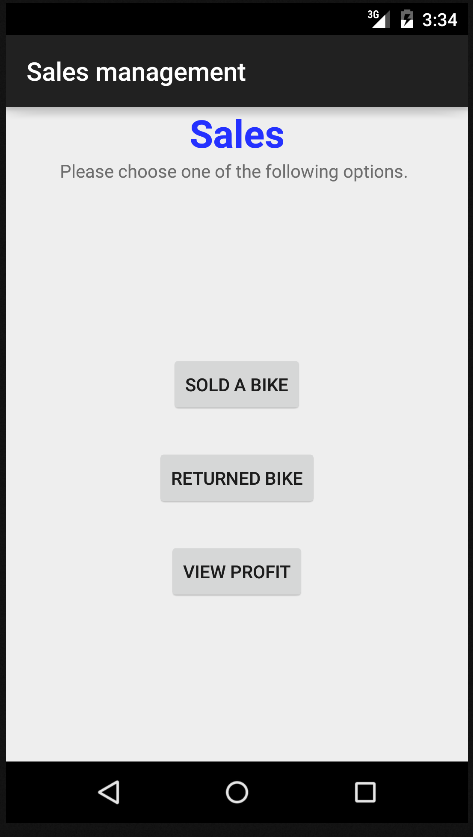
***Figure 8.*** This page will come up when the “Repairs/Tasks” button.

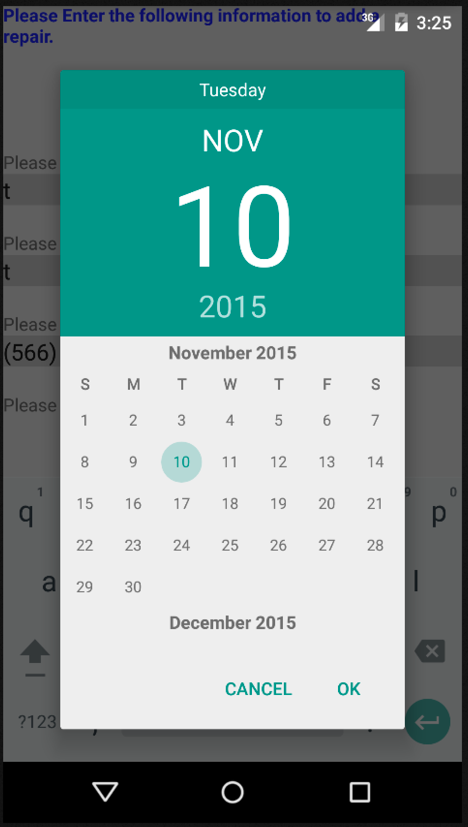
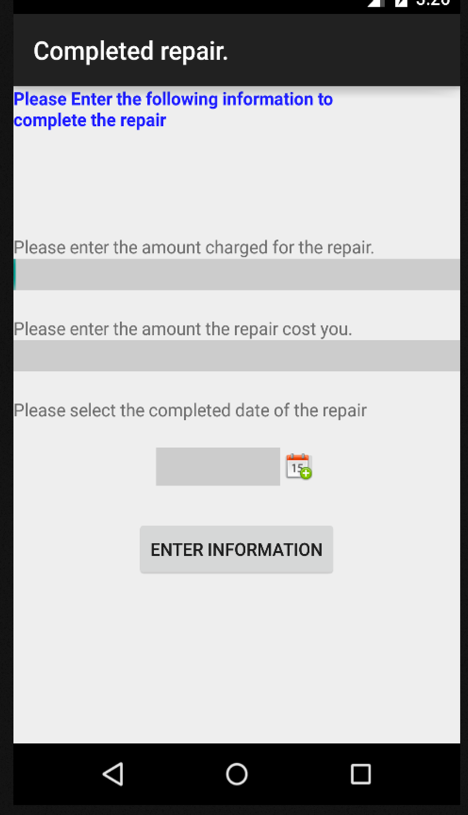
* + Add New Repair Button
    - * This button will allow the user to record a new repair.
* Completed/Update Repair button
  + This button will allow the user to calculate the profit of the repair when it’s finished and set the repair as complete.
* Delete repairs
  + This button will remove a repair from the list of current repairs.
* View active repairs
  + This button will display a list of active repairs on a new page
* View Pending repairs
  + This button will display a list of pending repairs on a new page.



***Figure 9.*** The “Add Repair” button will allow the user to add a repair. The button will display a new page with text boxes to input information about the bicycle and a dialog box to select a date. Once all the information is inputted a button at the bottom of the page will appear called “Enter Information”. This button will add all the information to the Repairs table and a bit 1 will be placed in the status column to signify that the bike is being repaired. The following will need to be inputted for Add Repair:

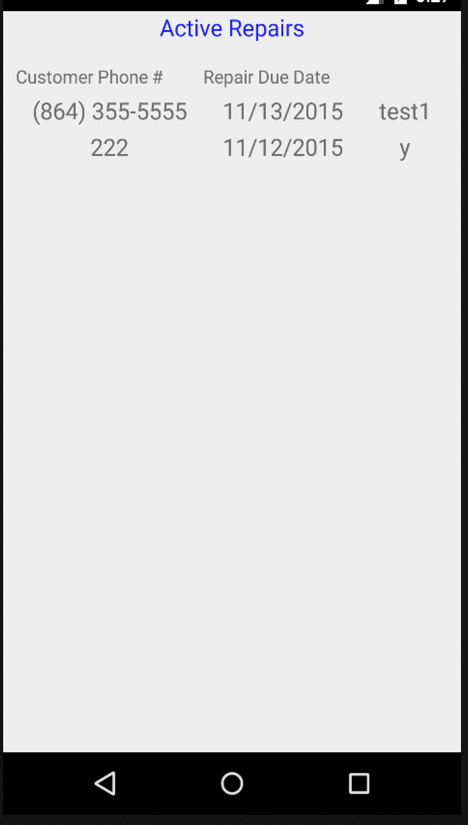
* + - * Customer name
      * Customer phone
      * Serial number of the bicycle
      * Due date for the repair.





**Figure 10.** The “Completed Repair” button will update information on the Repair table with the exception of the serial number and display a new page with the list of completed repairs which will be all of the repairs with a value of 1 in the status column of the repairs table. The following be inputted:

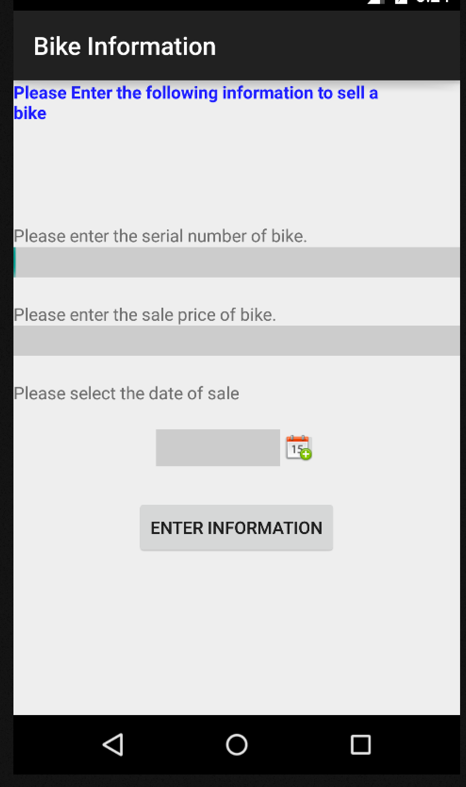
* + - * Total Repair cost to the owner
      * Amount charged to the customer



**Figure 11,** The “Active Repair” button will display a new page with the list of the active or current repairs which will be all of the repairs with a bit value of 1 in the status column of the repairs table.

### Hardware Interfaces

The application will store all of the data in a local database using SQLite. The database will consist of 3 primary tables. The application will automatically connect with the database and will be available for use while the app is running. Three tables will exist “Inventory” “Sales” and “Repairs”.

* The inventory table
* A repairs table
* A sales table

### Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>

### Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >

## Functional Requirements

The android application ProBikes will be an android phone application for the bike shop owner to use. The application will make it convenient for the owner to manage his business from his pocket.

* Once the application is launched the main page with the title Bikes will pop up and there will be three buttons to choose from. There will be “Manage Inventory”, “Sales”, and “Repairs” buttons on the main page.

### Home Page of Application

* + Manage Inventory Button
  + Sales Button
  + Repairs Button

### Inventory Management page

* The “Inventory Management” page will pop up once the “Manage Inventory” button is pressed from Section 3.2.1. This page will have three buttons listed below. This page is pictured in section 3.1.1 Figure 2.
  + Add Bicycle Button
  + Remove Bicycle Button
  + Display Inventory Button

### Sales page

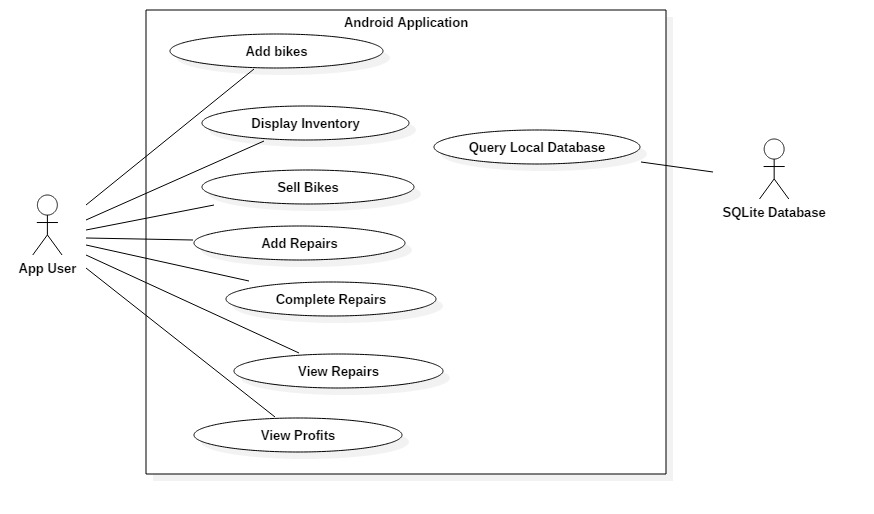
* This page will pop up once the button “Sales” Button is pressed from Section 3.2.1. This page will have three buttons listed below. A screen shot is pictured in Section 3.1.1 Figure 3.
  + Sell a Bike Button
  + View Profit
  + View Sold Bikes

### Repairs/Tasks page

* This page will come up when the “Repairs/Tasks” button is pressed from Section 3.2.1. This page will consist of six buttons listed below. Refer to Section 3.1.1 Figure 7.
  + Add Repair Button
* Complete Repair button
* Delete Repair
* View active repairs
* View Pending repairs
* View a Repair

## Behaviour Requirements

### Use Case View



# Other Non-functional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide at least 5 different performance requirements based on the information you collected from the client. For example you can say “1. Any transaction will not take more than 10 seconds, etc…>

## Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

* Provide at least 3 different safety requirements based on your interview with the client or, on your ABM related research, and again you need to be creative here.
* Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, & etc…Do not forget to include such attributes as the design for change. Please note that you need to include at least 2 quality attributes, but it is the mere minimum and it will not receive the full marks.>

Appendix A – Data Dictionary

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>