|  |
| --- |
| MIDDLE EAST TECHNICAL UNIVERSITY  INFORMATICS INSTITUTE  SM504 – TEAM SOFTWARE PROJECT |
| Ridder |
| Software Requirements Specification (SRS)  - Team D - |
| **Version 1.1** |

Contents

1. Introduction 4

1.1. Purpose 4

1.2. Scope 5

1.3. Definition, acronyms, and abbreviations 6

1.4. References 7

1.5. Overview 7

2. Overall Description 8

2.1. Product perspective 8

2.1.1. System interface 9

2.1.2. User interfaces 9

2.1.3. Hardware interfaces 9

2.1.4. Software interfaces 10

2.1.5. Communications interfaces 12

2.1.6. Memory constraints 12

2.1.6.1. Mobile Client 12

2.1.6.2. Server side 12

2.1.7. Operations 13

2.1.8. Site adaption requirements 13

2.2. Product functions 13

2.2.1. Manage Content Sources 13

2.2.2. Manage Categories 13

2.2.3. Obtain Card Content 14

2.2.4. Manage Statistics 14

2.2.5. View Users 14

2.2.6. Manage Cards 14

2.2.7. Manage Pocket Account 14

2.3. User characteristics 14

2.4. Constraints 15

2.4.1. Regulatory policies 15

2.4.2. Hardware limitations 15

2.4.3. Interfaces to other applications 15

2.4.4. Parallel operation 15

2.4.5. Audit functions 16

2.4.6. Control functions 16

2.4.6.1. Input control 16

2.4.6.2. Field control 16

2.4.7. Higher-order language requirements 16

2.4.8. Signal handshake protocols 16

2.4.9. Reliability requirements 16

2.4.10. Criticality of the application 17

2.4.11. Safety and security considerations 17

2.5. Assumptions and dependencies 17

2.6. Apportioning of requirements 17

3. Specific requirements 19

3.1. External interface requirements 19

3.1.1. Mobile Application Main Page 19

3.1.2. Settings Page 22

3.1.3. Login Page 24

3.1.4. Sources Page 25

3.1.5. Categories Page 26

3.1.6. Statistics Page 27

3.1.7. Users Page 27

3.2. Functional requirements 29

3.2.1. User Classes 29

3.2.1.1. Mobile User ……………………………………………………………………………………………….27

3.2.1.2. Administrator 36

3.3. Performance requirements 48

3.4. Logical database Requirements 48

3.5. Design constraints 48

3.6. Software System Attributes 49

APPENDIX A – List of Tables 52

APPENDIX B – Data Model Diagram 53

APPENDIX C – Traceability Information 54

|  |
| --- |
|  |

**Document control**

|  |  |  |
| --- | --- | --- |
| Version Number | Date | Description |
| 1.0 | March 22nd, 2015 | Baseline SRS |
| 1.1 | April 1st , 2015 | Revised SRS |

# Introduction

This Software Requirement Specification document is prepared for Ridder, which is basically a system that finds articles from a variety of sources assigned to different categories, displays the title and short summary of said articles via a mobile phone interface in card style and lets user like and save the article to her Pocket account or dislike and discard the article by swiping the card to the right or left. This document is written in accordance with the IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications [1].

## Purpose

In the last years, along with the increasing rate of mobile device usage, reading habits of the people has also evolved. As a result of this evolution, many people started to read magazine and blog articles on different topics through their mobile devices. There are some services that enhance this reading experience and one of them is Pocket [2]. This application serves as both mobile application and a web application and lets users to save articles and videos to read or watch later.

This project is undertaken for providing a mobile solution which will have a server and client side application where client (mobile) application displays a title and a short summary of an article on a card and which will let smartphone users to swipe the card to right or left to like and save it to their Pocket account or dislike the card.

A system called Ridder will be developed and will eliminate the need and effort for browsing many different resources and many different articles and posts over Web since the server side application will scan and find the categorized articles published in the last pre-determined time and push their title and respective short summaries to the mobile application. Users will be able to find and select among many interesting articles in a very short time to read them later in their Pocket account.

This document aims to present detailed description of the requirements expected from the product. This document will also be used in order to establish the basis for agreement between the supplier (Project Team D) and customers (Instructor and Teaching Assistant) on final product.

## Scope

Ridder will be developed in order to provide targeted articles to the users based on their preferences. The system will allow the user to select categories of the articles and keep the user activities in order to decide which articles to be sent to the specific user. For administration of the resources and categories, the system will allow the administrative users to manage system entities.

Objectives of Ridder:

* To eliminate the need of a user to browse many different sources for finding stuff to read
* To display titles and short summaries for contemporary articles from different user-preferred categories one by one in a card style on a smartphone screen
* To provide an efficient and intuitive user experience for saving articles to Pocket account for a later reading

Ridder will contain:

* A mobile application for the mobile users to see and swipe article cards
* A web based application for the administration of the system
* A virtual web server where the web based application will be deployed
* A database to store the user-based and statistical data

Ridder will allow:

1. To manage article sources by the system administrator
2. To determine a category for the article sources by the system administrator
3. Explore and find the new articles on the sources periodically
4. Determine a title and a short summary for the articles from resources
5. To select preferred categories by the mobile application user
6. Display title and short summary of an article on a card according to the preferred selection of the user
7. To swipe the card right by user to save the full article to Pocket account
8. To swipe the card left by user to be “Not interested” in the article
9. To connect the Pocket account to the mobile application account
10. Keep the statistical information as user-based and generally for the articles:
    1. Keep the viewed, left-swiped and right-swiped numbers for each specific card for each day
    2. Keep the right-swiped numbers for each specific source for each specific user for whole time usage
    3. Keep the total swipe numbers (right or left) for each specific user on a daily basis
    4. Keep the total swipe numbers for whole users
11. Learn the user preferences better by examining user actions against the cards shown on the mobile application
    1. If a user right-swipes a card from a specific source at more than 50% ratio, show the cards from that source at a 75% ratio at a new session.
    2. Find the top right-swiped 3 cards for each category in the last 6 hours and sort them at the beginning of a new card stack of a new session for the upcoming 6 hours.
12. Mark a left-swiped card article as “not to be shown again” card for that specific user

Ridder will not allow:

1. Keep the saved article on the device storage or on a database managed by the server
2. Provide full text of the articles
3. Open the related page for the article on the device browser
4. To rate the articles on a scale by the user

## Definition, acronyms, and abbreviations

IEEE The Institute of Electrical and Electronics Engineers

SRS Software Requirements Specification

URL Uniform Resource Locator

HTML Hypertext Markup Language

OS Operating System

DB Database

## References

[1] IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications. (ISBN 0-7381-0332-2).

[2] Pocket, <https://getpocket.com/>

## Overview

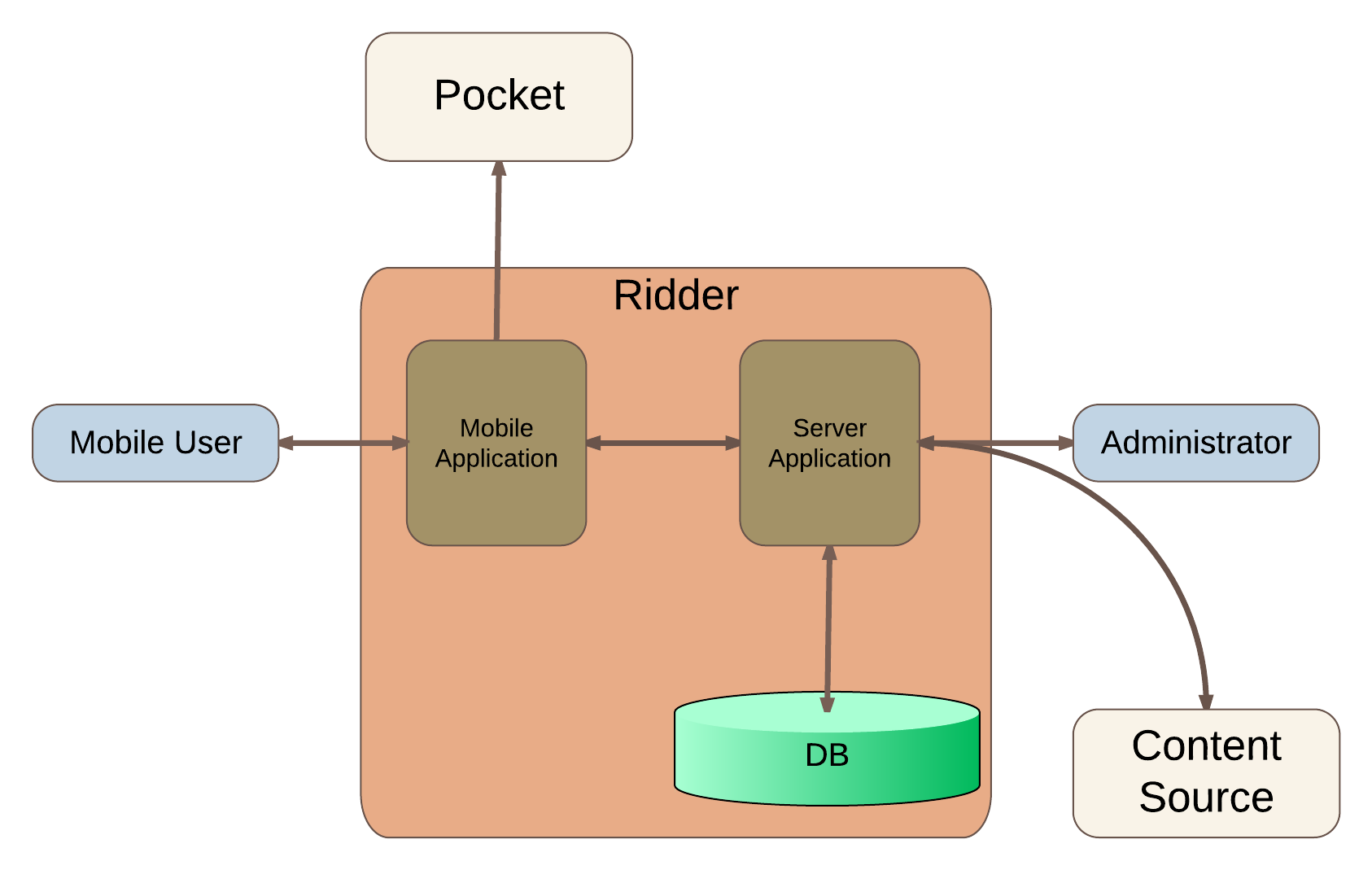
IEEE Recommended Practice for Software Requirements Specifications [1] is utilized in creating this SRS document. The rest of the document provides the information of Product Perspective, Product Functions, Specific Requirements, User Characteristics, Constraints, Assumptions and Dependencies.

# Overall Description

This section of the SRS document describes the general factors that affect the Ridder and its requirements. It does not state specific requirements, but provides a background for them. Detailed specifications of the requirements are presented in Section 3 of this document.

## Product perspective

Figure 2-1 shows the user types and major components of the Ridder. User types which will interact with the system are Mobile User and Administrator. Major components of the system are Ridder mobile application, Ridder administrative web application and RDBMS. Pocket will be used to save preferred articles by the user.



**Figure 2-1 Major components of the Ridder**



### System interface

There shall be an interface between Ridder and Pocket to send the selected article information and mobile user account details to Pocket.

There shall be another interface between Ridder and Content Source to pull the content such as title, text to summarize and a related image of the article.

### **User interfaces**

Below are the main features of the user interfaces:

* English will be the default language of the user interfaces.
* A standard theme (colors, fonts, backgrounds) and structure (page layout) will be used for mobile application user interface according to Android Material Design Guidelines.
* A standard theme (colors, fonts, backgrounds) and structure (page layout) will be used for the server application.
* Success or failure messages for each operation will be provided on each related user interface.
* Success and failure messages shall include clear success or failure words along with the related process name to each type of user, and comprehensible in terms of which parts of the system completed the action or caused the error.

### **Hardware interfaces**

There are no hardware interface requirements for the Ridder. System will be accessible via Android mobile devices for the mobile users and any device with standard web browser for the administrator.

### **Software interfaces**

* Pocket API (<http://getpocket.com/developer/docs/overview>) shall be an interface between Ridder and Pocket to send the selected article information and mobile user account details to Pocket.
* There shall be another interface between Ridder and Content Source, which would be specific to Content type (for example RSS or source code parsing style or another content API), to pull the content such as title, text to summarize and a related image of the article.

For user software to interact with Ridder;

* Administrators can access the server application user interface by Google Chrome (v41.0 or above) and Mozilla Firefox (v36.0 or above) web browsers running on Windows OS, Mac OS or Ubuntu.
* Mobile users can access the mobile application user interface by Android OS (v4.4 or above).
* The details of these products are given in Table 2-1, Table 2-2, Table 2-3, Table 2-4 and Table 2-5.

|  |  |
| --- | --- |
| Name | Windows |
| Mnemonic | OS |
| Specification number | - |
| Version number | 7 (or later) |
| Source | Microsoft Corp. |
| Reference | <http://windows.microsoft.com/en-US/windows/home> |

**Table 2-1 Windows OS**

|  |  |
| --- | --- |
| Name | Mac OS X |
| Mnemonic | Mavericks |
| Specification number | - |
| Version number | 10.9 (or later) |
| Source | Apple Inc. |
| Reference | <http://www.apple.com/macosx/> |

**Table 2-2 Mac OS X**

|  |  |
| --- | --- |
| Name | Mozilla Firefox |
| Mnemonic | - |
| Specification number | - |
| Version number | 36 |
| Source | Mozilla Corp. |
| Reference | <http://www.mozilla.org/en-US/firefox/new/> |

**Table 2-3 Mozilla Firefox**

|  |  |
| --- | --- |
| Name | Google Chrome |
| Mnemonic | - |
| Specification number | - |
| Version number | 41 |
| Source | Google Inc. |
| Reference | <http://www.google.com/chrome> |

**Table 2-4 Google Chrome**

|  |  |
| --- | --- |
| Name | Android |
| Mnemonic | Kitkat |
| Specification number | - |
| Version number | 4.4 |
| Source | Google Inc. |
| Reference | https://www.android.com/ |

**Table 2-5 Android**

### **Communications interfaces**

* The communication protocol to be used for data transmission between mobile application and server application is the Transmission Control Protocol (TCP) and the Internet Protocol (IP).
* The communication between mobile application and Pocket will occur compatible with Hypertext Transfer Protocol Secure (HTTPS).
* The communication between server application and content sources will occur compatible with Hypertext Transfer Protocol (HTTP).
* OS will carry out communication between server application and DB since both of them will be on the same virtual machine.

### **Memory constraints**

### Mobile Client

The mobile client, which is a smartphone, must be capable of running Android 4.4 or above and should have at least 1 GB of memory. The mobile client should also have a 100 MB of free storage for the content cache on the cards and Pocket account information. User based data, category data and content sources related data will be kept on the DB which resides on the virtual server machine.

### Server side

The server side of the Ridder will require at least 2 GB of memory and 20GB of storage to serve the system, run the server OS and store the DB.

### **Operations**

* The mode of the system is only operational mode.
* In operational mode, system shall work as expected. User operations will be performed during the operational mode of the system.
* There will be no mode for unattended operations like back-up and recovery since the system does not require these operations.

### **Site adaptation requirements**

* For the server application, no specific installation will be required since the server application will be a web-based application.
* For the mobile application, installation of an Android package will be required on the Android OS running on smartphone, through Google Play Store or directly from the device storage.

## Product functions

Below are the summary of the major functions that the software shall perform.



### Manage Content Sources

Managing content sources shall have the following features: creating/editing/deleting content sources.

### Manage Categories

Managing categories shall have the following features: creating/editing/deleting categories.

### Obtain Card Content

Obtaining the card content should have the following features: accessing the content/obtaining title for the content/generating summary for the content/obtaining a related thumbnail/assigning an 8-digit ID starting from 00000001 and incrementing for the content/recording the contentID-sourceID-categoryID as correlated.

### Manage Statistics

Viewing statistics shall have the following features: generating/viewing staticstics.

### View Users

Viewing sources shall have the following features: viewing user and user related data.

### Manage Cards

Managing cards shall have the following features: viewing/saving/discarding the cards.

### Manage Pocket Account

Managing Pocket account shall have the following features: adding the Pocket account username and password/logging out from Pocket.

## User characteristics

* The intended users of the Ridder are grouped in 2 user types, which are namely “Mobile User” and “Administrator”.
* No specific educational background is required for any user type.
* The basic knowledge of how to use a smartphone running Android is adequate for the mobile user. The basic knowledge of how to use computer, its peripheral devices like mouse or keyboard and Internet browsers is adequate for the administrator.
* The certain specific requirements are later specified in section 3 of this SRS document since they are strongly related with use cases.

## Constraints



### **Regulatory policies**

No specific policy will be used for Ridder.

### **Hardware limitations**

Memory requirements of mobile client and server side are stated in Section 2.1.6.

The mobile client should be a smartphone, which is capable of running Android 4.4 or above.

The server side hardware should be a virtual machine, which is capable of running Ubuntu 14.04 LTS.

### **Interfaces to other applications**

Ridder shall provide an interface for authorizing and sending related URL of selected content card to Pocket. When the mobile user swipes a card to the right, the URL for the content on the card will be sent to the Pocket and stored in the user’s Pocket account.

### **Parallel operation**

There are no parallel operations in the context of Ridder.

### **Audit functions**

There are no audit functions related to Ridder.

### **Control functions**

### Input control

Ridder shall check the input types and validity ranges in each user interfaces whenever it gets a new record or update request.

### Field control

Ridder shall check availability of essential fields in each user interfaces whenever a new record, update or query operation is requested.

System shall also check the input values whether they are valid or not according to their predefined user types.

### **Higher**-**order language requirements**

There are no higher-order language requirements.

### **Signal handshake protocols**

There are no constraints on signal handshake protocols for Ridder.

### **Reliability requirements**

Since the system will determine and store statistical information for user and content-based data, any loss of data cannot be permitted.

### **Criticality of the application**

Ridder Software is a non-critical application.

### **Safety and security considerations**

* Server application in the system is only accessible after passing its logon interface. So, all administrators need to have their own username and password to access the system.
* Mobile users will input their Pocket account credentials to the mobile application. So, this account credentials shall be transmitted to Pocket through Pocket API and without any man-in-the-middle security defect.
* User-based statistical information shall be accessible only when an administrator passes the server application’s logon interface.
* Any user-based statistical information should be kept anonymously, only with unique user ID.

## Assumptions and dependencies

AD 1. The mobile user will have a Pocket account in order to be able to use Ridder.

AD 2. Pocket services will be up and running.

AD 3. It is assumed that users have the necessary software mentioned in 2.1.4 on their systems to access the Ridder.

AD 4. Physical security of the Ridder server is under the responsibility of the virtual cloud server provider.

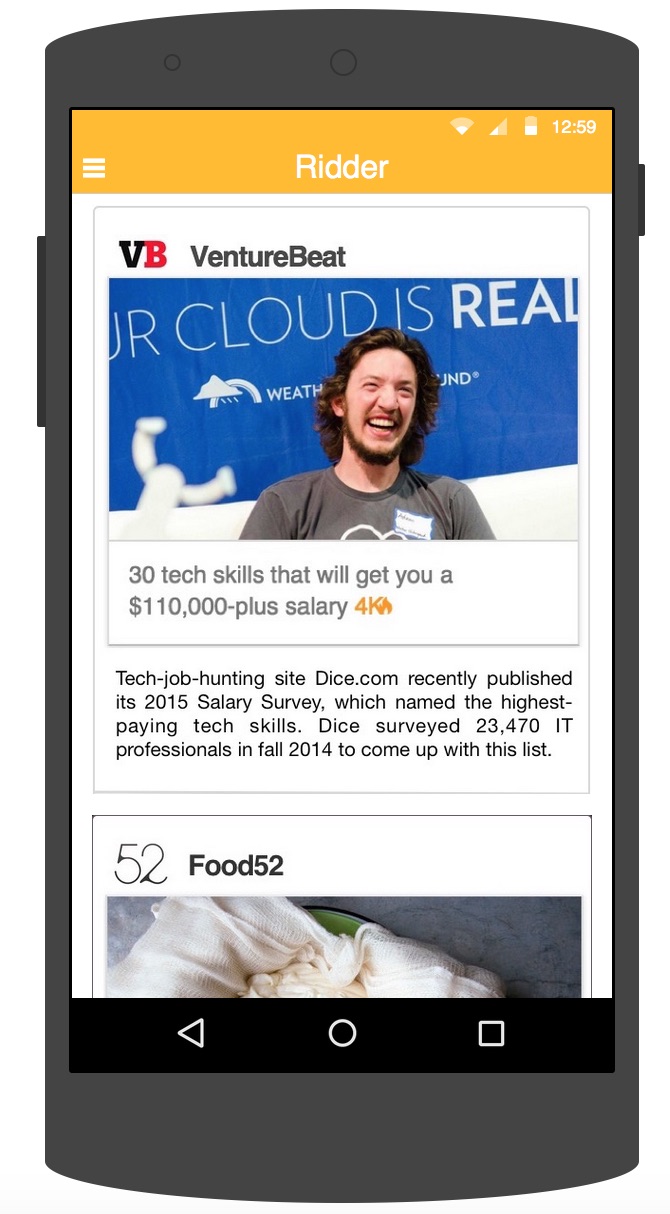
## Apportioning of requirements

Ridder has no requirements that may be delayed until future versions of the development.

# Specific requirements

## External interface requirements

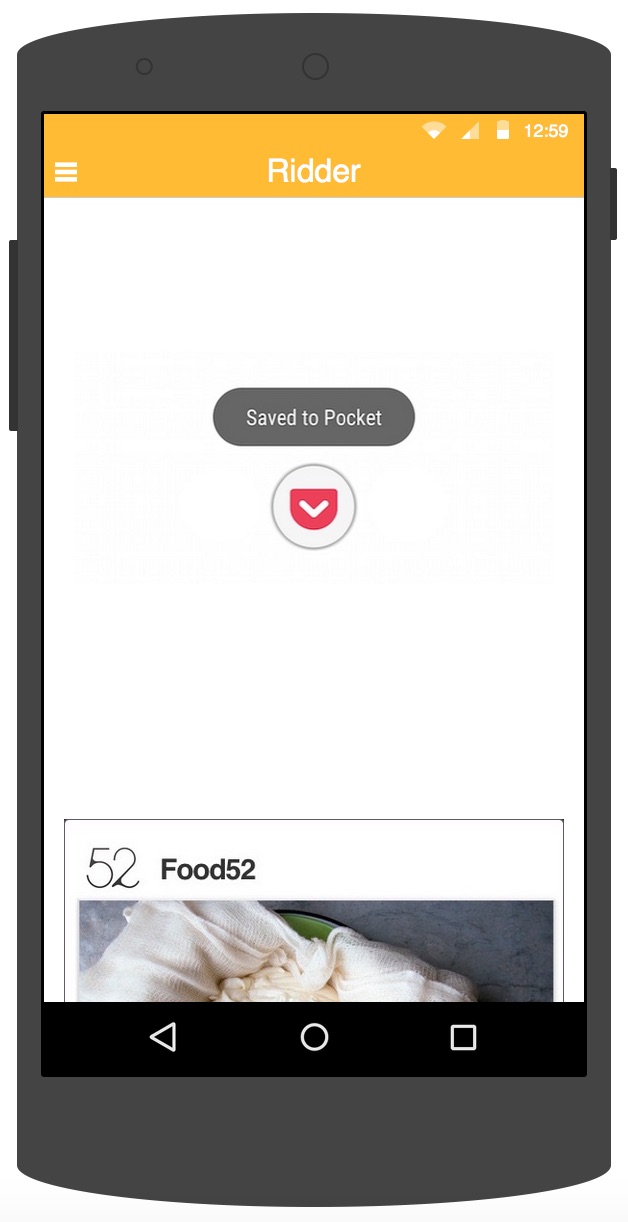
### Mobile Application Main Page



**Figure 3-1 Mobile Application Main Page**

Mobile application will have a main page that displays articles as content on the card and one by one to the mobile user. User can swipe the content card right or left to indicate his interest to the article.

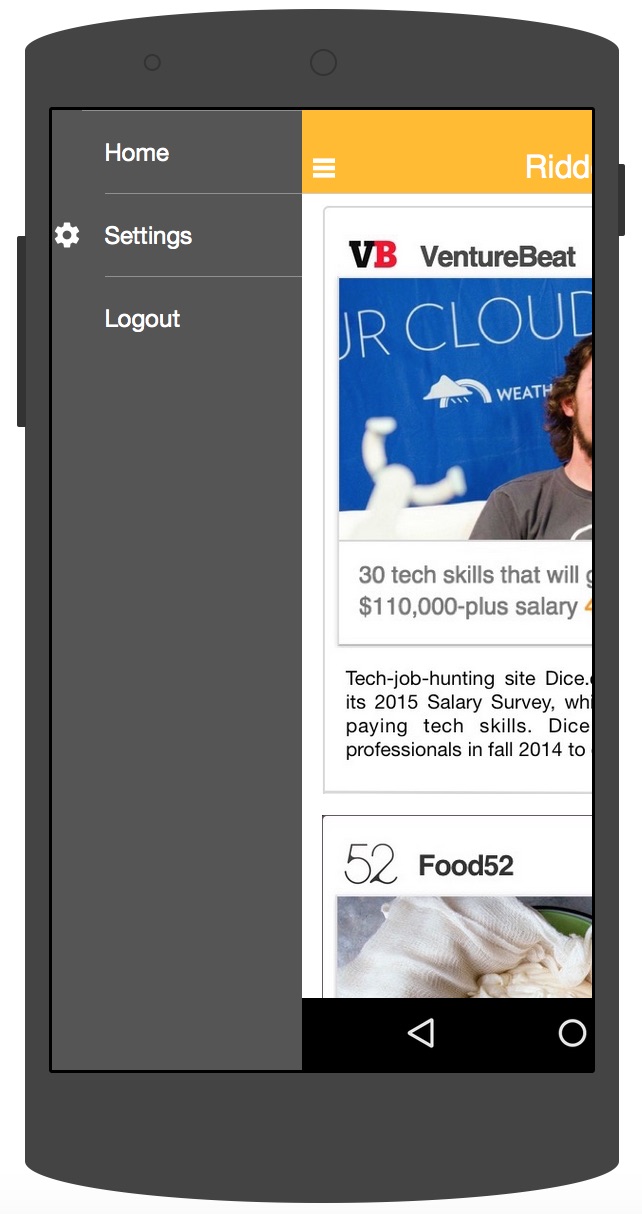
When a user swipes the content card to right, the URL of the article content on the card is saved to mobile user’s Pocket account by the system and a message regarding this process is shown to the user on mobile application user interface.



**Figure 3-2 Mobile Application Page when a Content Card is Swiped to Right**

The system will show the next card when “Saved to Pocket” message is displayed to user for 400 ms.

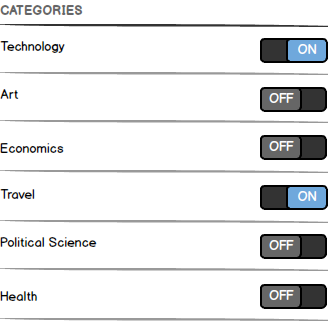
The mobile application main page displays the menu navigation through its UI element on the left corner of the screen, which means menu navigation according to Android Material Design Guidelines.



**Figure 3-3 Mobile Application Main Page showing the Menu Navigation**

|  |  |
| --- | --- |
| **UI Control** | **Description** |
| Title (Text Control) | Shows the title of the content. Read-only 3 lines text control. |
| Summary (Text Control) | Shows the summary of the article content. Read-only 10 lines of text control. |
| Right Swipe (Gesture) | Adds the article to the user’s Pocket account. Sends this information along with the content ID to the server application. |
| Left Swipe (Gesture) | Marks the article as “not interested” and sends this information to the server application. |

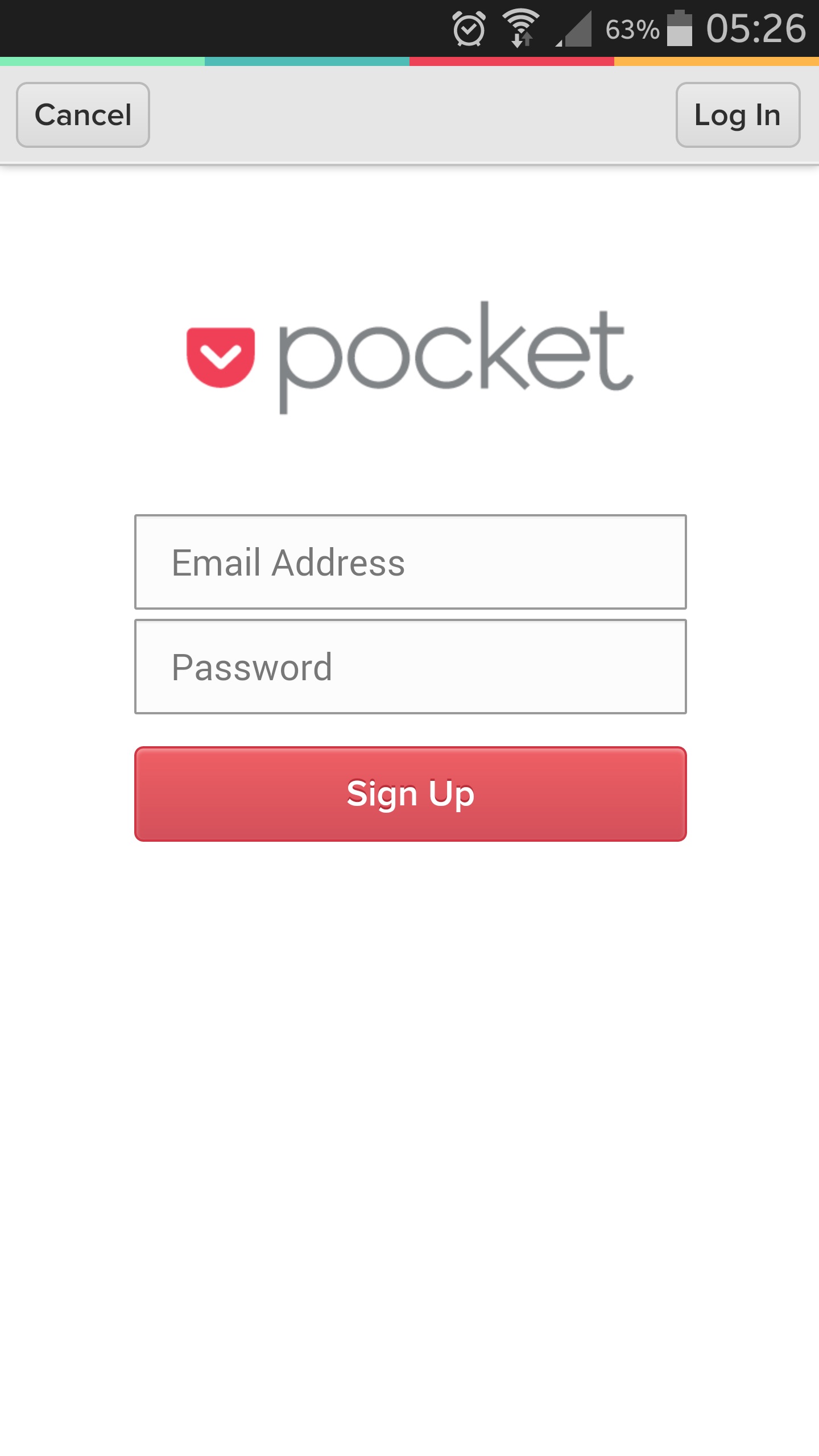
### Settings Page



**Figure 3-4 Mobile Application Settings Page**

This page will allow the user to switch the categories on or off based on his interest. The articles to be listed on the main page will be retrieved from the server based on the enabled categories of the user.

This page will also allow the user to manage the Pocket account by showing “Add Pocket account” or “Logout” button along with the logged-in Pocket account name regarding the log-in condition of the mobile user.



**Figure 3-5 Pocket Account Login/SignUp Page**

|  |  |
| --- | --- |
| **UI Control** | **Description** |
| Category (Text Control) | Shows the category name. Read-only single line text control. |
| Switch (Switch) | Enables/disables the category. |
| Add Pocket Account (Button) | Opens the Pocket account login or sign-up page if no Pocket account is logged-in. |
| Pocket Account Name (Text Control) | Shows the Pocket account name that is logged-in. |
| Logout (Button) | Logs out the Pocket account. |

### Login Page

The system will have a login page for administrator authentication. Any other pages will be accessible after successfully passing this login page. The login page is accessible by an IP address or a domain name configured on the server.

|  |  |
| --- | --- |
| **UI Element** | **Description** |
| Username (TextBox) | Min 4 max 10 alphanumeric characters |
| Password (PasswordBox) | Min 4 max 10 alphanumeric characters |
| Login (Button) | Submits to form for login |

Login page leads the administrator to “Categories Page with a sidebar menu for navigation” after a successful login.

### Sources Page

This page will be the entry point for administrators after login. The system will have this sources page, which will let the administrator manage the content sources. The page will display the existing content sources along with their source IDs, name, categories, main source URLs in a table style where each row of the table is for one source and columns are for source parameters and actions.

The page will also have the links for adding new source/updating an existing source/deleting a source.

|  |  |
| --- | --- |
| **UI Element** | **Description** |
| Add New (Link) | Open “Add New Source” form |
| Update (Link) | Open “Update Source” form for related source |
| Delete (Link) | Deletes the source. |
| Source List (DataGrid) | Contains the list of the sources and parameters. |

While adding a new source or updating an existing source, the following UI elements should exist on the form as stated in the description.

|  |  |
| --- | --- |
| **UI Element** | **Description** |
| Source Name (TextBox) | Min 1 max 100 alphanumeric characters |
| Source URL (TextBox) | Min 1 max 100 alphanumeric characters |
| Category  (Dropdown List) | Category list is displayed which allows only one selection. |
| Submit (Button) | Clicked to add new source/update a source and return to sources page |
| Cancel (Button) | Returns to sources page without any operations. |

When a source is added, the system will assign a 6 digit numerical ID for it and save the source with this ID. First ID for the sources is 000001 and followings are incremental.

### Categories Page

The system will have categories page, which will let the administrator manage the categories. The page will display the existing categories along with their category IDs and name.

The page will also have the links for adding new category/updating an existing category/deleting a category.

|  |  |
| --- | --- |
| **UI Element** | **Description** |
| Add New (Link) | Open “Add New Category” form |
| Update (Link) | Open “Update Category” form for related category |
| Delete (Link) | Deletes the category. |
| Source List (DataGrid) | Contains the list of the categories. |

While adding a new category or updating an existing category, the following UI elements should exist on the form as stated in the description.

|  |  |
| --- | --- |
| **UI Element** | **Description** |
| Category Name (TextBox) | Min 1 max 30 alphanumeric characters |
| Submit (Button) | Clicked to add new category/update a category and return to categories page |
| Cancel (Button) | Returns to categories page without any operations. |

When a category is added, the system will assign a 4 digit numerical ID for it and save the category with this ID. First ID for the categories is 0001 and followings are incremental.

### Statistics Page

This page will allow the administrator to display the statistics. The page will display

1. Viewed, left-swiped and right-swiped numbers for each specific content card for each day
2. The total swipe numbers for whole users
3. The right-swiped numbers for each specific source for each specific mobile user for whole time usage
4. The total swipe numbers (right or left) for each specific mobile user on a daily basis

The statistics page can be accessed for general statistics (for a and b) or for user-based statistics (c and d) from the user list on the Users Page described below.

### Users Page

This page will allow the administrator to display the mobile user list and UI elements to show user related data.

User ID is identical the Unique Device ID (UDID) of the smartphone that mobile user is using.

|  |  |
| --- | --- |
| **UI Element** | **Description** |
| Show Stats (Link) | Opens the viewed vs. liked stats for each category and sources for the user. |
| User List (DataGrid) | Contains the list of the users and user IDs. |

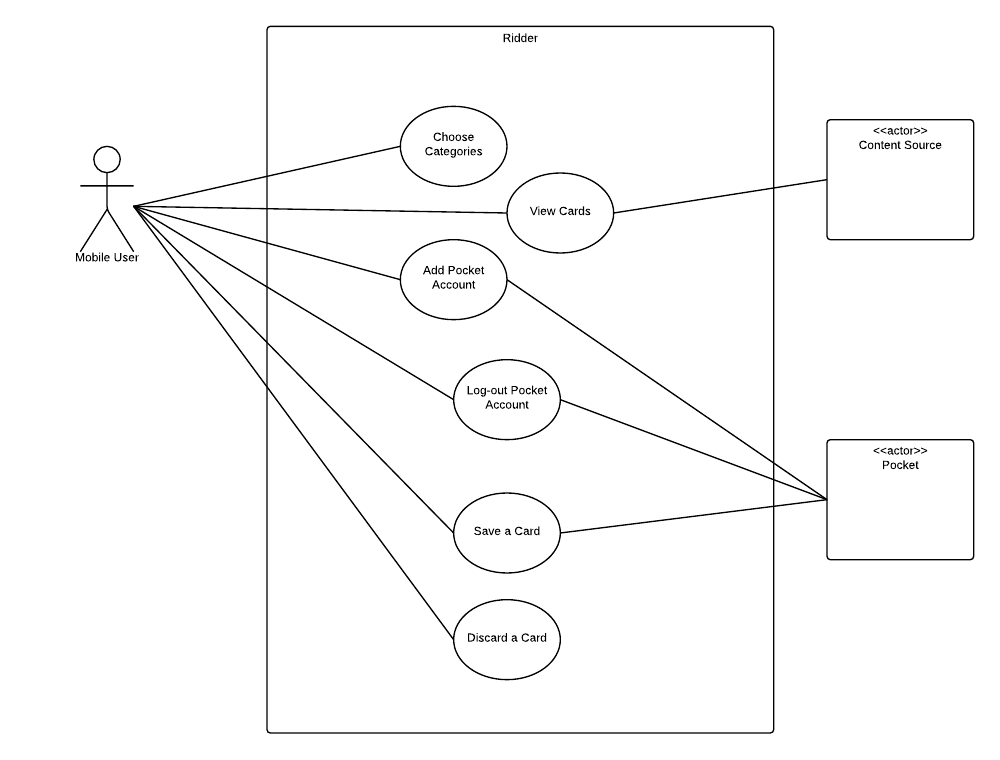


## Functional requirements

### User Classes

* + - 1. **Mobile User**





|  |  |
| --- | --- |
| ***UC-MBL-01: Choose Categories*** | |
| Use Case UC-MBL-01 | Choose Categories |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Mobile User |
| Stakeholders and Interests | Mobile user wants to choose the categories according to his interests so that the content cards will be from the sources in selected categories |
| Preconditions | Smartphone has an active Internet connection. |
| Success Guarantee | User’s category preferences are saved to DB. |
| Main Success Scenario | 1. User wants to choose categories. 2. User opens the settings page. 3. User switches the button on or off for the selected categories. 4. User determines the time interval for content time. 5. User taps back button. 6. Ridder saves the category preferences of mobile user in DB. 7. Ridder saves the time interval preference of the user. 8. Ridder displays a confirmation message. |
| Extensions | 1. User does not change any category switch.    1. Ridder does not change the settings.    2. Ridder displays the home screen. 2. User does not change time interval info.    1. Ridder does not change the settings.    2. Ridder displays the home screen. |
| Special Requirements | There are no special requirements |

|  |  |
| --- | --- |
| ***UC-MBL-02: View Cards*** | |
| Use Case UC-MBL-02 | View Cards |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Mobile User |
| Stakeholders and Interests | Mobile user wants to view the content cards. |
| Preconditions | Smartphone has an active Internet connection.  User has chosen the preferred categories (Use Case UC-MBL-01). |
| Success Guarantee | Content cards are shown to mobile user. |
| Main Success Scenario | 1. User wants to view content cards. 2. Ridder determines the sources from user’s preferred categories. 3. Ridder determines the time interval preference of the user. 4. Ridder determines 100 content from the determined sources. 5. Ridder sorts the content according to user statistics. 6. Ridder sorts the content according to general statistics. 7. Ridder extracts title for the sorted contents. 8. Ridder determines the summary for sorted content. 9. Ridder determines a thumbnail image for the sorted content. 10. Ridder displays the cards one by one to mobile user. |
| Extensions | 1. Ridder cannot determine a thumbnail image for the content.   1.1. Ridder displays card with grey background instead of a thumbnail image. |
| Special Requirements | Ridder sorts the content according to following rule:  a. If a user has right-swiped (saved) a card from a specific source at more than 50% ratio, display the cards from that source at a 75% ratio at a new session, all at the beginning.  b. Find the top right-swiped (saved) 3 cards for each category in the last 6 hours and sort them at the beginning of a new card stack of a new session for the upcoming 6 hours.  c. Rule b has priority over Rule a.  d. If no rule applies, display the cards in random order. |

|  |  |
| --- | --- |
| ***UC-MBL-03: Add Pocket Account*** | |
| Use Case UC-MBL-03 | Add Pocket Account |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Mobile User |
| Stakeholders and Interests | Mobile user wants to add Pocket account so that he can save the content cards to Pocket account. |
| Preconditions | Smartphone has an active Internet connection. |
| Success Guarantee | Mobile user’s Pocket account is authorized within Ridder. |
| Main Success Scenario | 1. User wants to add Pocket account. 2. User opens the settings page. 3. User taps Add Pocket Account button. 4. Ridder displays Pocket account username and password fields. 5. User enters Pocket account credentials. 6. Ridder sends credentials to Pocket over API. 7. Ridder receives a success message from Pocket through API. 8. Ridder displays success message to mobile user. 9. Ridder displays settings page. |
| Extensions | 1. User enters an incorrect username/password.    1. Ridder receives a failure message from Pocket over API.    2. Ridder displays a failure message to mobile user.    3. Ridder displays Pocket account username and password fields |
| Special Requirements | There are no special requirements |

|  |  |
| --- | --- |
| ***UC-MBL-04: Log-out Pocket Account*** | |
| Use Case UC-MBL-04 | Log-out Pocket Account |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Mobile User |
| Stakeholders and Interests | Mobile user wants to log-out (remove) the Pocket account. |
| Preconditions | Smartphone has an active Internet connection.  Mobile user’s Pocket account is authenticated within Ridder. (UCL-MBL-03) |
| Success Guarantee | Mobile user’s Pocket account is de-authorized within Ridder. |
| Main Success Scenario | 1. User wants to logout the Pocket account. 2. User opens the settings page. 3. User taps Log-out button. 4. Ridder sends log-out request to Pocket over API 5. Ridder receives a success message from Pocket through API. 6. Ridder displays log-out success message to mobile user. 7. Ridder displays settings page. |
| Extensions | - |
| Special Requirements | There are no special requirements |

|  |  |
| --- | --- |
| ***UC-MBL-05: Save a Card*** | |
| Use Case UC-MBL-05 | Save a Card |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Mobile User |
| Stakeholders and Interests | Mobile user wants to save a card to Pocket account. |
| Preconditions | Smartphone has an active Internet connection.  User has chosen the preferred categories (Use Case UC-MBL-01).  Cards are displayed to user. (Use Case UC-MBL-02).  User has added a Pocket account. (Use Case UC-MBL-03). |
| Success Guarantee | Right-swiped card is saved to Pocket account. |
| Main Success Scenario | 1. User wants to save the content card. 2. User right-swipes the card. 3. Ridder determines the URL of the content card. 4. Ridder sends authorization information to Pocket over API. 5. Ridder sends URL of the content card to Pocket to be saved. 6. Ridder receives a success message from Pocket. 7. Ridder displays success message “Saved to Pocket” to mobile user. 8. Ridder updates statistics for this saved card and mobile user. 9. Ridder displays the next card to mobile user. |
| Extensions | 1. Ridder receives a failure message from Pocket.    1. Ridder displays a failure message to mobile user.    2. Ridder displays the same card to mobile user. 2. Ridder receives a failure message from Pocket.    1. Ridder displays “Add a Pocket account” message to mobile user. 3. Ridder cannot update the statistics for saved card and mobile user.    1. Ridder does not display any message to user. |
| Special Requirements | Ridder updates the statistics as follows:  For each right-swiped card, the content and the source gets +1 point for the user.  For each right-swiped card, the content gets +1 point for all users. |

|  |  |
| --- | --- |
| ***UC-MBL-06: Discard a Card*** | |
| Use Case UC-MBL-06 | Discard a Card |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Mobile User |
| Stakeholders and Interests | Mobile user wants to discard a card. |
| Preconditions | Smartphone has an active Internet connection.  User has chosen the preferred categories (Use Case UC-MBL-01).  Cards are displayed to user. (Use Case UC-MBL-02).  User has added a Pocket account. (Use Case UC-MBL-03). |
| Success Guarantee | Left-swiped card is discarded and next card is shown. |
| Main Success Scenario | 1. User wants to discard the content card. 2. User left-swipes the card. 3. Ridder updates the statistics. 4. Ridder determines the card not to be shown again to same mobile user. 5. Ridder displays the next card to mobile user. |
| Extensions | 1. There is no next card for the mobile user.   1.1. Ridder displays a message stating that mobile user has displayed all cards for him. |
| Special Requirements | Ridder updates the statistics as follows:  For each left-swiped card, the content and the source gets (-1) point for the user.  For each left-swiped card, the content gets (-1) point for all users. |

### Administrator



Manage Categories (UC-ADM-02) has 4 use cases as View Categories, Add Category, Update Category and Delete Category.

Manage Sources (UC-ADM-03) has 4 use cases as View Sources, Add Source, Update Source and Delete Source.

|  |  |
| --- | --- |
| ***UC-ADM-01: Login*** | |
| Use Case UC-ADM-01 | Login |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to log in to Ridder server application successfully |
| Preconditions | Administrator is defined in Ridder: Administrator exists in the system with a valid username and password combination |
| Success Guarantee | Administrator is successfully logged in Ridder. |
| Main Success Scenario | 1. Administrator wants to log in to Ridder server application. 2. Ridder shows a login screen. 3. Administrator enters his/her username and password. 4. Ridder verifies entered information. 5. Ridder logs administrator in. |
| Extensions | 1. Administrator forgets his/her password.    1. Administrator changes the password manually in DB.    2. Use case continues with Main Success Scenario Step 2. 2. Username and/or password entered by the Administrator are not valid.    1. Ridder displays “Invalid username and/or password” error message.    2. Use case continues with Main Success Scenario Step 2. |
| Special Requirements | There are no special requirements |

|  |  |
| --- | --- |
| ***UC-ADM-02-1: View Categories*** | |
| Use Case UC-ADM-02-1 | View Categories |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to view the categories |
| Preconditions | Administrator is logged in. |
| Success Guarantee | Category list is displayed successfully. |
| Main Success Scenario | 1. Administrator wants to display categories. 2. Ridder displays category list to admin. |
| Extensions | 1. There are no categories defined in Ridder.    1. Ridder displays “No category yet” informative message. |
| Special Requirements | There are no special requirements |

|  |  |
| --- | --- |
| ***UC-ADM-02-2: Add Category*** | |
| Use Case UC-ADM-02-2 | Add Category |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to create a new category |
| Preconditions | Administrator is logged in. |
| Success Guarantee | A new category is added successfully. |
| Main Success Scenario | 1. Administrator wants to add a new category. 2. Ridder displays the new category form. 3. Administrator enters the name of the category. 4. Ridder assigns an ID to category. 5. Ridder records the new category. 6. Ridder displays a success message. 7. Ridder displays the category list. |
| Extensions | 1. There is an existing category with the same name defined in Ridder.    1. Ridder displays “Existing category” error message.    2. Use case continues with Main Success Scenario Step 2. |
| Special Requirements | There are no special requirements |

|  |  |
| --- | --- |
| ***UC-ADM-02-3: Update Category*** | |
| Use Case UC-ADM-02-3 | Update Category |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to update a category |
| Preconditions | Administrator is logged in.  Category list is displayed successfully. (Use Case UC-ADM-02-1) |
| Success Guarantee | An existing category is updated successfully. |
| Main Success Scenario | 1. Administrator wants to update a category. 2. Ridder displays the category information form. 3. Administrator changes the name of the category. 4. Ridder records the new name. 5. Ridder displays a success message. 6. Ridder displays the category list. |
| Extensions | 1. There is an existing category with the same name defined in Ridder.    1. Ridder displays “Existing category” error message.    2. Use case continues with Main Success Scenario Step 2. |
| Special Requirements | There are no special requirements |

|  |  |
| --- | --- |
| ***UC-ADM-02-4: Delete Category*** | |
| Use Case UC-ADM-02-4 | Delete Category |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to delete a category |
| Preconditions | Administrator is logged in.  Category list is displayed successfully. (Use Case UC-ADM-02-1) |
| Success Guarantee | An existing category is deleted successfully. |
| Main Success Scenario | 1. Administrator wants to delete a category. 2. Ridder displays the category information. 3. Administrator clicks Delete button. 4. Ridder deletes the record for the category. 5. Ridder displays a success message. 6. Ridder displays the category list. |
| Extensions | - |
| Special Requirements | Ridder shall display the “Some sources might be assigned to no categories” informative message. |

|  |  |
| --- | --- |
| ***UC-ADM-03-1: View Sources*** | |
| Use Case UC-ADM-03-1 | View Sources |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to view the sources |
| Preconditions | Administrator is logged in. |
| Success Guarantee | Source list is displayed successfully. |
| Main Success Scenario | 1. Administrator wants to display sources. 2. Ridder displays source list to admin. |
| Extensions | 1. There are no sources defined in Ridder.    1. Ridder displays “No sources yet” informative message. |
| Special Requirements | Ridder shall display source list as stated in 3.1.4. |

|  |  |
| --- | --- |
| ***UC-ADM-03-2: Add Source*** | |
| Use Case UC-ADM-03-2 | Add Source |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to create a new source |
| Preconditions | Administrator is logged in. |
| Success Guarantee | A new source is added successfully. |
| Main Success Scenario | 1. Administrator wants to add a new source. 2. Ridder displays the new source form. 3. Administrator enters the name of the source. 4. Administrator enters the URL of the source. 5. Administrator chooses a category for the source. 6. Ridder assigns an ID to source. 7. Ridder records the new source. 8. Ridder displays a success message. 9. Ridder displays the source list. |
| Extensions | 1. There is an existing source with the same URL defined in Ridder.    1. Ridder displays “Existing source” error message.    2. Use case continues with Main Success Scenario Step 2. |
| Special Requirements | There are no special requirements |

|  |  |
| --- | --- |
| ***UC-ADM-03-3: Update Source*** | |
| Use Case UC-ADM-03-3 | Update Source |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to update a source |
| Preconditions | Administrator is logged in.  Source list is displayed successfully. (Use Case UC-ADM-03-1) |
| Success Guarantee | An existing source is updated successfully. |
| Main Success Scenario | 1. Administrator wants to update a source. 2. Ridder displays the source information form. 3. Administrator changes the attributes of the source. 4. Ridder records the source with new attributes. 5. Ridder displays a success message. 6. Ridder displays the source list. |
| Extensions | 1. There is an existing source with the same URL defined in Ridder.    1. Ridder displays “Existing source” error message.    2. Use case continues with Main Success Scenario Step 2. |
| Special Requirements | There are no special requirements |

|  |  |
| --- | --- |
| ***UC-ADM-03-4: Delete Source*** | |
| Use Case UC-ADM-02-4 | Delete Source |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to delete a source |
| Preconditions | Administrator is logged in.  Source list is displayed successfully. (Use Case UC-ADM-03-1) |
| Success Guarantee | An existing source is deleted successfully. |
| Main Success Scenario | 1. Administrator wants to delete a source. 2. Ridder displays the source information. 3. Administrator clicks Delete button. 4. Ridder deletes the record for the source. 5. Ridder deletes the content from the deleted source. 6. Ridder displays a success message. 7. Ridder displays the source list. |
| Extensions | - |
| Special Requirements | - |

|  |  |
| --- | --- |
| ***UC-ADM-04: View General Statistics*** | |
| Use Case UC-ADM-04 | View General Statistics |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to view the general statistics |
| Preconditions | Administrator is logged in. |
| Success Guarantee | General statistics are displayed successfully. |
| Main Success Scenario | 1. Administrator wants to display general statistics. 2. Ridder displays statistics to admin. |
| Extensions | - |
| Special Requirements | Ridder shall display general statistics as stated in 3.1.6. |

|  |  |
| --- | --- |
| ***UC-ADM-05: View Users*** | |
| Use Case UC-ADM-05 | View Users |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to view the mobile users |
| Preconditions | Administrator is logged in. |
| Success Guarantee | Mobile user list is displayed successfully. |
| Main Success Scenario | 1. Administrator wants to display user list. 2. Ridder displays user list to Admin. |
| Extensions | 1. There are no users in Ridder.    1. Ridder displays “No users yet” informative message. |
| Special Requirements | Ridder shall display source list as stated in 3.1.7. |

|  |  |
| --- | --- |
| ***UC-ADM-06: View User Statistics*** | |
| Use Case UC-ADM-04 | View User Statistics |
| Scope | Ridder |
| Level | User-goal |
| Primary Actor | Administrator |
| Stakeholders and Interests | Administrator wants to view the user statistics |
| Preconditions | Administrator is logged in.  User list is displayed successfully. (Use Case UC-ADM-05) |
| Success Guarantee | User statistics are displayed successfully. |
| Main Success Scenario | 1. Administrator wants to display user statistics. 2. Administrator clicks user statistics link on user list. 3. Ridder displays user statistics to Admin. |
| Extensions | - |
| Special Requirements | Ridder shall display user statistics as stated in 3.1.6. |

## Performance requirements

* The time between request and response for every web request (request time limit) should be less than 10 seconds.
* At the same time system should be usable by 100 users within the request time limits.
* System should be at ready to save article state under 10 seconds after saving an article to pocket.
* For any kind of error, system should warn the user less than 27 seconds about the problem.

## Logical database Requirements

Refer to Appendix B for the Logical Data Model Diagram of the system.

## Design constraints

Object-oriented programming approach shall be used in software development for maintainability, flexibility, and extensibility of the code in the future.

For any kind of design pattern usage, Addy Osmani’s design pattern templates will be applied to be able to keep object-oriented approach in Javascript codes. See <http://addyosmani.com/resources/essentialjsdesignpatterns/book/> for further information.

* + 1. Standards compliance
* The design of the software shall be made in accordance with UML 2.1 notations.
* JavaScript code style should follow Google JavaScript style guide at <https://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml>
* HTML and CSS style should follow Google HTML/CSS style guide at <https://google-styleguide.googlecode.com/svn/trunk/htmlcssguide.xml>
* Android application user interfaces should follow the Android Material Design Guidelines stated at <http://www.google.com/design/spec/material-design/introduction.html>

## Software System Attributes

* + 1. Reliability
* Ridder’s Pocket integration shall be functional as long as Pocket interface is available.
* The system shall be up for 98% time of the year.
* All content cards swiped right or left shall function as instructed.
  + 1. Availability
* In case of any crash on the system server, system shall be available within 6 hours.
  + 1. Security
* Administrator authentication with a user name and password is needed to use the server application.
* For administrator password, MD5 password encryption shall be used.
* Input to password field shall be masked.
* A warning message shall be shown to administrators in case of unsuccessful login attempts.
* The system shall not allow adding a new administrator through a user interface.
* Adding a new administrator is allowed by adding credentials manually after accessing DB on the server.
  + 1. Maintainability
* Ridder will allow for extensions and updates for the new features and bug fixes.
* For extension, inner and external coupling should be less than 3 for each class and/or module, i.e. Low coupling-high cohesion philosophy should be considered for the system.
* SRS and SDD documents shall be up to date during and after the development of software in order to provide maintainability.
  + 1. Portability
* Ridder shall be accessible via all smartphones running Android 4.4 or above (for the mobile application part of the system).
* Ridder shall be accessible via operating systems (Windows OS, Mac OS, and Unix/Linux), which are capable of running internet-based web browsers mentioned at 2.1.4 (for the server application part of the system).

# APPENDIX A – List of Tables

Table 2-1 Windows 7 10

Table 2-2 Mac OS X 10

Table 2-3 Mozilla Firefox 11

Table 2-4 Google Chrome 11

Table 2-5 Android 11

# APPENDIX B – Data Model Diagram



# APPENDIX C – Traceability Information

**Project Charter Scope Definiton vs. Use Cases**

* 1 – UC-ADM-03
* 2 – UC-ADM-03-2 and UC-ADM-02
* 3 – UC-MBL-02
* 4 – UC-MBL-02
* 5 – UC-MBL-01
* 6 – UC-MBL-02
* 7 – UC-MBL-05
* 8 – UC-MBL-06
* 9 – UC-MBL-03
* 10 – UC-ADM-04 and UC-ADM-06
* 11 – UC-MBL-02
* 12 – UC-MBL-02