

Gifter Requirements

5.1 Introduction

Gifter's system is mobile-centric and includes an iOS client written in Apple's new programming language, Swift. This client performs HTTP RESTful requests to the Java API hosted on a server. From there, the Java API interfaces with a Neo4j database running on a different server. This flow of data can be seen below in Figure 5.1a.

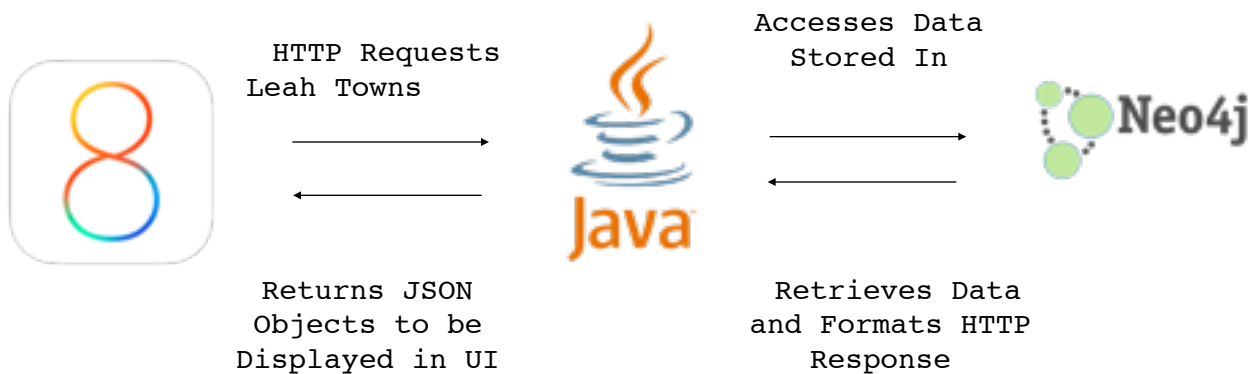


Figure 5.1a

The following sections are organized as follows: Section 5.2 contains the functional requirements of the overall system, Section 5.3 contains performance requirements with regards to the above components, and section 5.4 contains the environment in which each of these components will live.

5.2 Functional Requirements

5.2.1 Graphical User Interface

The Graphical User Interface (GUI) designed in Xcode will facilitate the user to interact with the application.

5.2.1.1 The GUI shall provide a tabular system to access the various views listed in section 5.2.2. The tabular system will resemble the same one that can be found in Apple's phone application in iOS.

5.2.2 Views

5.2.2.1 The GUI shall provide a login view that allows the user to keep his/her information secure and private.

5.2.2.1.1 Logging in will require a username and password.

5.2.2.1.2 The username must be unique within the Gifter database and will preferably be an email address.

5.2.2.1.3 The password will be more stringent than usual social applications. Passwords must be a minimum of 8 characters in length and require at least one number and one capital letter.

5.2.2.1.4 A user will be able to attempt to login in 5 times after which there will be a hold placed on the username. The hold will last 20 minutes in which the user will not be able to login with that specific username.

5.2.2.1.5 If the user has forgotten their username or password they can click a button that will send an email to the username specified to optionally reset the password.

5.2.2.2 The GUI shall provide a view that lists the items that the user has favorited.

5.2.2.2.1 This view will contain a button that will allow the user to search for other items to favorite via key search terms. To view the item's details the user can click on the item and will be taken to a item detail view.

5.2.2.2.2 In order to remove an item the user can swipe right on an item and a delete button will appear. A modal confirmation will pop up if the delete button is clicked.

5.2.2.3 The GUI shall provide a item detail view that displays the item's details as provided by Amazon's API.

5.2.2.3.1 This view will also have a back button at the top, left header of the view to take the user back to the item list view.

5.2.2.4 The GUI shall provide a friends list view that will allow the user to view their friends they have searched for and added after approval of the friend.

5.2.2.4.1 There will be a button on the friends list view that allows the user to search for other friends to add with a button to friend request the desired user. Upon clicking the "Request to Friend" button a confirmation modal of the request being successfully sent or not

will pop up. If it failed, there will be a button on the modal to optionally retry.

5.2.2.4.2 Tapping on a friend will show the specified user's favorites item view. This view will function the same as the user's own item list view.

5.2.2.4.3 This view will also allow the user to swipe right on a user in the list. This will provide a optional delete button from the friends list in the exact same fashion as the item view's swipe right function.

5.2.2.5 The GUI shall provide a third tabular view that will contain all of the user's editable information with fields that include, but are not limited to, the following: username, email, password, phone number, credit card and billing information.

5.3 Performance Requirements

5.3.1 User Info Pull Down Time

5.3.1.1

At the launch of the application, Gifter shall perform a query to pull down the user's information from the API's data storage. The user's item list, friend list, and personal information shall be pulled down in 10 seconds or less (pending connection quality).

5.3.2 Search for Other Users Response Time

5.3.2.1

From the moment the user types in a username, email, or phone number into the search box the system should take 10 seconds or less to retrieve the possible matches. This will not include the time it takes to display the information either.

5.3.3 Search for Items to Favorite

The search for other items to favorite will be based upon the performance of Amazon's inventory API. Given their amount of resources and the performance seen in their iOS applications, the retrieval of possible search matches shall be 8 seconds or less.

5.3.4 Purchasing an Item

When a user purchases an item, he/she knows that their own money and credit card information is on the line. In order to achieve a feeling of security and sufficient performance, the process of conducting a transaction shall take no more than 8 seconds.

5.4 Environment Requirements

5.4.1 Hardware Requirements

Category	Requirement
Phone/Tablet	Apple Mobile Device
Server	Virtual Machine running a linux distribution, 16 Gigs of ram, 4 Ghz Quad-Core Processor, 1 Tb of hard drive space

5.4.2 Software Requirements

Category	Requirement
iOS	Version 7 or higher
API Language	Java
Mobile Device Language	Swift
Additional Libraries for API	Jersey
Additional Libraries for iOS	Alamofire, SwiftyJSON
Mobile Development Environment	Xcode 6
API Development Environment	Eclipse Luna