Group King 17
Software Requirements Specification
For CollegeBook

Version 1.0

Revision History

Date	Version	Description	Author
25/3/18	1.0	Highlight important features in the CollegeBook web application	Nicholas Kane

Table of Contents

1. Introduction

1.1 Purpose
1.2 Scope
1.3 Definitions, Acronyms and Abbreviations
1.4 References
1.5 Overview

2. Overall Description

2.1 Use-Case Model Survey
2.2 Assumptions and Dependencies

3. Specific Requirements

3.2 Supplementary Requirements

4. Supporting Information

Software Requirements Specification

1. Introduction

The goal of this project is to develop our skills in web development. Additionally, we are trying to gain a better understanding of how to collaborate in teams using object-oriented programing. We hope to continue to use the programing skills in this class to complete more projects in the future, and further or passions within the web development space.

In this document, we will…

1.1 Purpose

CollegeBook is intended to be a social media application that is used to post and share status updates between friends through the internet. It gives students the ability to share content online quickly and with ease; the web application is modeled after Facebook. By creating a design similar to Facebook, we will be able to have many opportunities to communicate and develop on a back-end server, as well as design a front-end interface that users can easily navigate.

1.2 Scope

This project has a short scope. The first version of our project will be due on April 26. This will have the intial layouts with all our pages, with few errors between backend queries and a basic front-end design. The second version will be due at the end of this class (early May). This version will have a more refined front-end design, and additional backend features will be added if time permits it.

1.3 Definitions, Acronyms and Abbreviations

The project will be developed using a local server tool called WAMP for windows, and MAMP for Macs. This acronym WAMP/MAMP stands for Windows/Mac, Apache, MySQL, and PHP. SQL stands for Structed Query Language. PHP used to stand for Personal Home Page, but now stands for the recursive acronym PHP: Hypertext Preprocessor.

1.4 References

In section 4 (Supporting Information) of this document, we have supporting files that detail our Use-Case Diagrams, Use-Case Reports, and Sequence Diagrams. With these images along with the specific requirements details, it will be easy for a developer to start writing code for this system.

1.5 Overview

Section 2 of this document highlights the major classes used within the system, and what user stories are used and associated with those classes. Section 3 goes into more detail by tying these user stories to use-case reports and sequence diagrams; this will make it possible for programmers to view this section and understand what they are developing. Section 4 has images attached, which can be used and referenced by programmers to build the necessary classes and functions for the user stories and system to be developed

2. Overall Description

Our development will depend on each team member's ability to pick up on and implement the object-oriented programing model within a PHP and MySQL context. To create all our webpages, we currently plan to develop content under 2 general classes, and 5 classes for a specific web interface type. The 2 general classes are Database and Page classes; the 5 other classes are Home, Login, Group, Profile, and Message classes. The Database and Page classes enable backend data communication, and will be needed in order for the 5 other classes to generate dynamic content for our webpage interfaces

2.1 Use-Case Model Survey

In this document, we have currently created 3 use case diagrams. One diagram is for the home page, which includes a user as the actor and 3 user stories: post status, like or comment on post, and search in search bar for people and groups. Having the option to search for people and groups will provide the user access to view more webpages, while the post and like/comment options let the user interact with each webpage. Another diagram is for the profile page, which also includes a user as the actor and 3 user stories: access group list, access friend list, and access all posts by user. From this diagram, we give the user another way to interface he can use to access webpages of his groups and friends. The final diagram is for the login in page with a user actor and 3 user stories: create personal account, login to account, and create group account.

2.2 Assumptions and Dependencies

One major assumption is that users will want to view and interact with different groups and friends on the website. Another assumption is that there will be multiple users with accounts and groups who are signed into the website simultaneously; if there are not users who are already made prior to the site's launch, then the site will not have has much content, and consequently be less successful

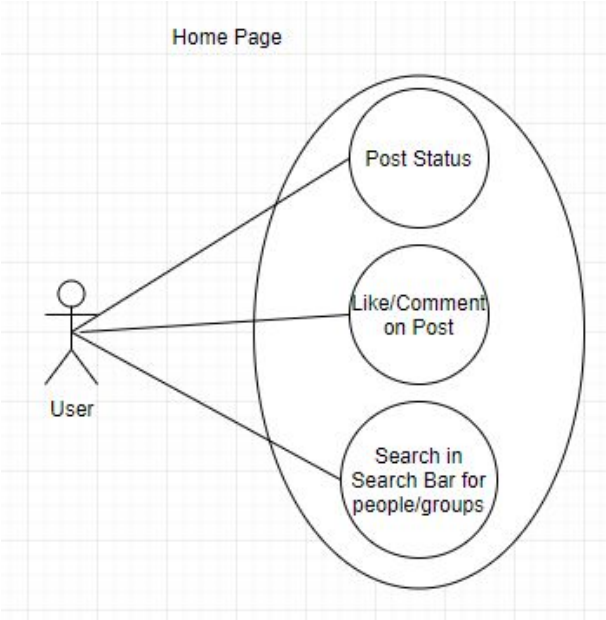
3. Specific Requirements

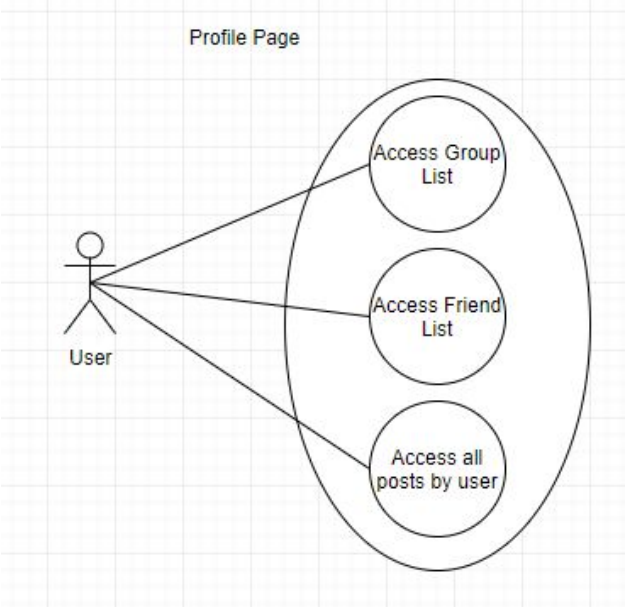
3.2 Supplementary Requirements

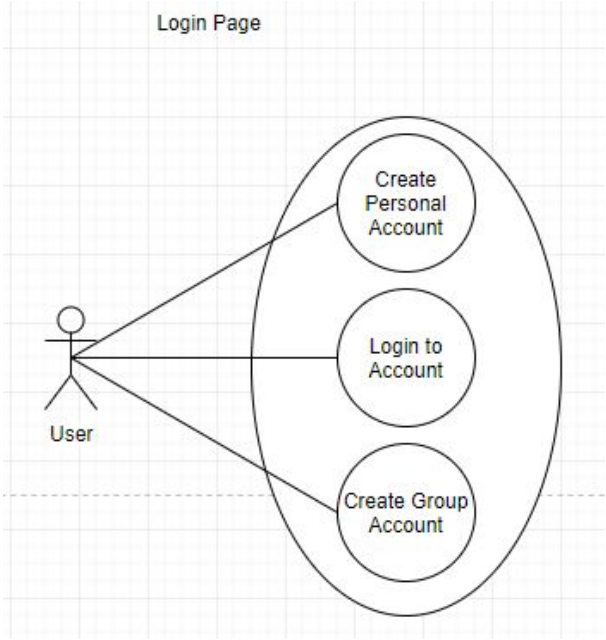
For important user stories in our project, we used sequence diagrams to highlight what interactions needed to happen in the frontend and backend of our system. One diagram was for creating a new account, and another was for logging in. Both of these use a main function to get into the database, and the database performs internal processing to determine if the account or login was performed successfully. For creating a post, the text of the post is sent to the database, and that is saved into the system then redisplayed to the user. For adding a friend or group, the username is processed by the databased and the user is then alerted that their request has been sent. For accepting a friend or group request, a Boolean is sent that the request has been accepted and the database then adds that person or group to user's list; the system also displays a message saying the request has been successfully accepted.

4. Supporting Information

4.1 Use-Case Diagrams



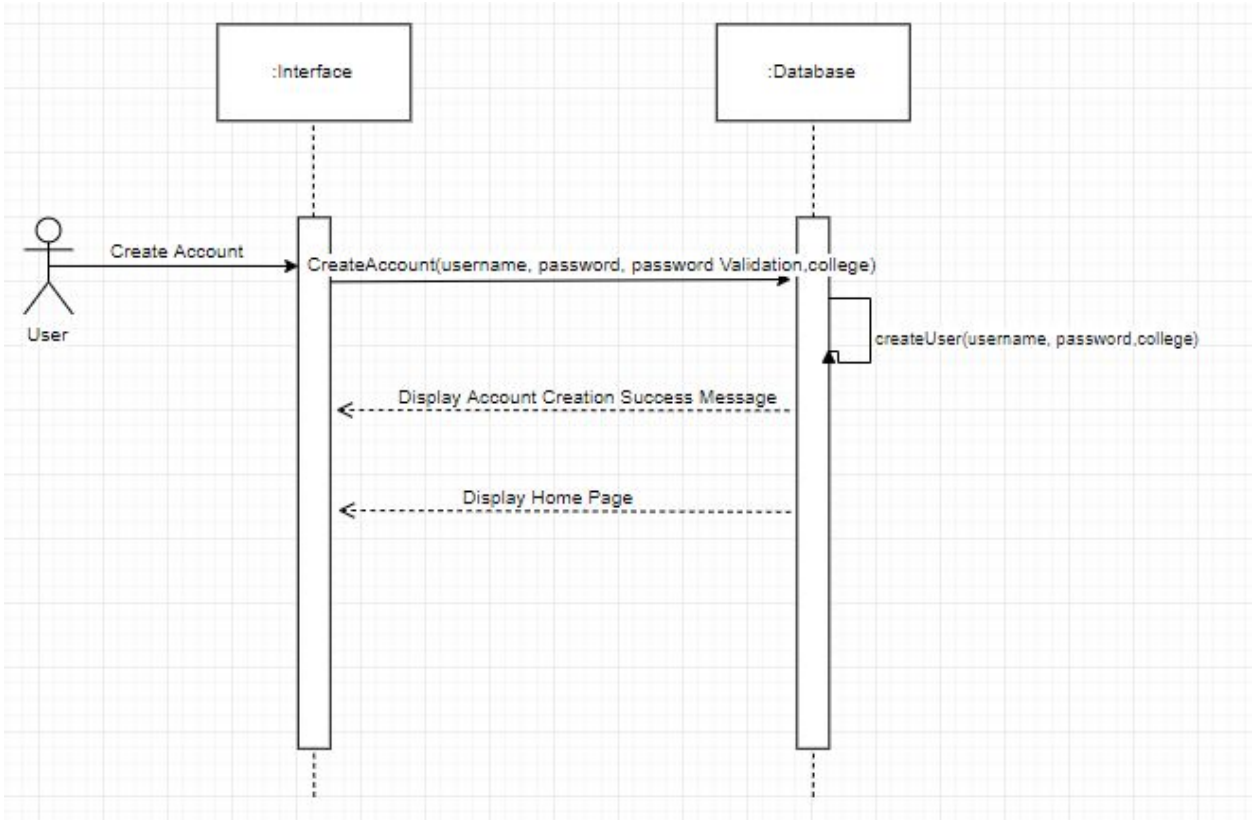




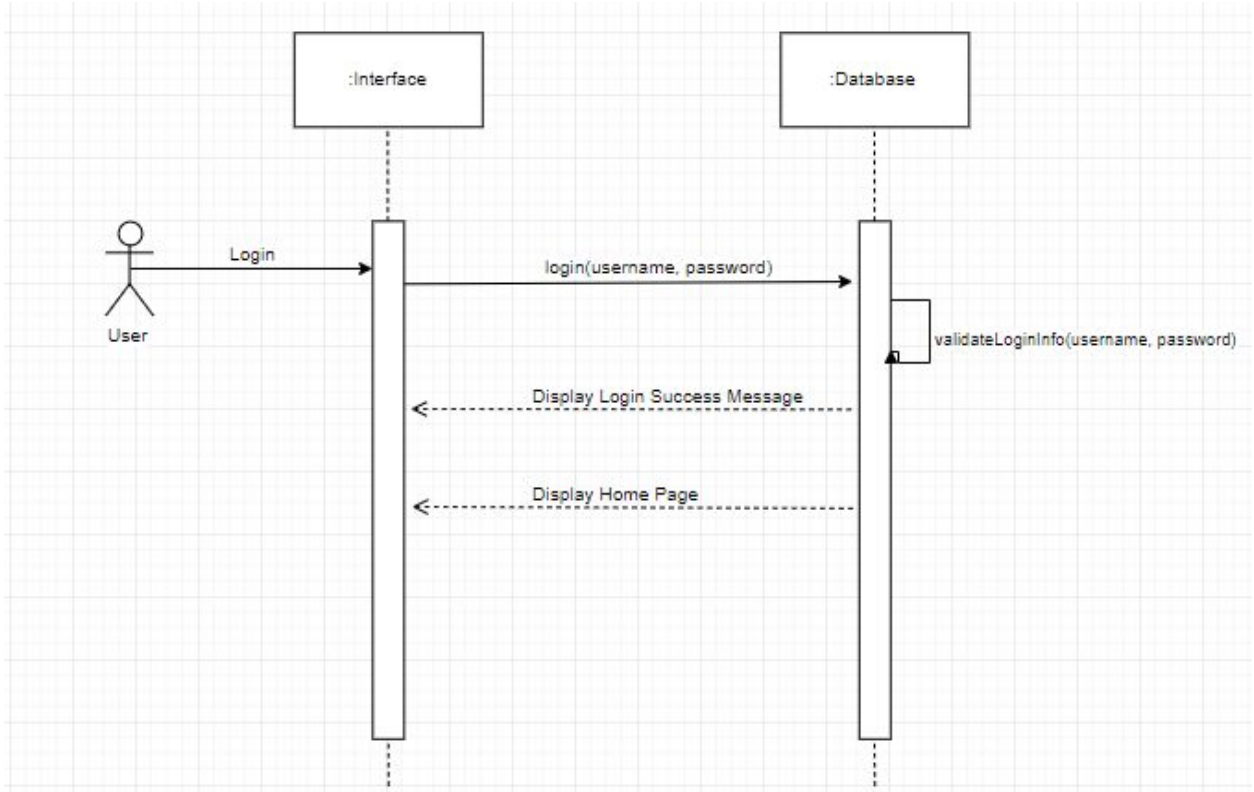
4.2 Use-Case Reports

4.3 Sequence Diagrams

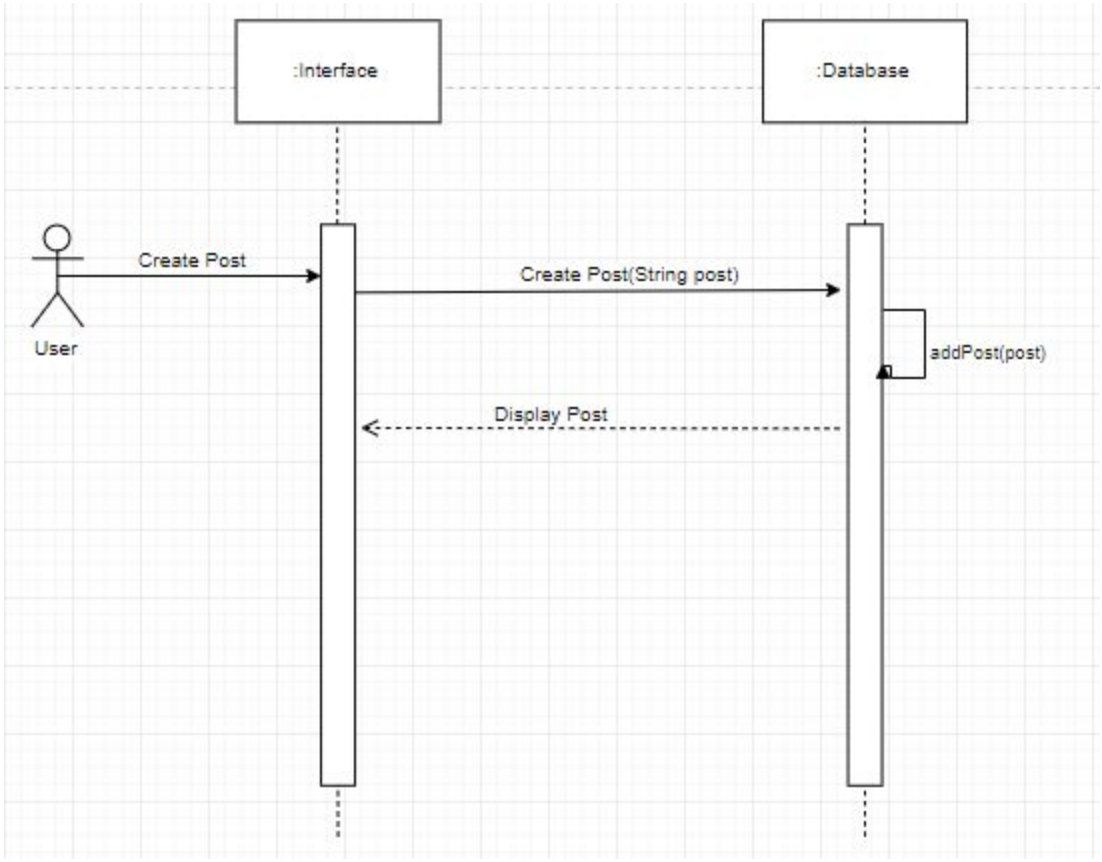
Create Account:


:Interface
:Database
User
Create Account
CreateAccount(username, password, password Validation,college)
createUser(username, password,college)
Display Account Creation Success Message
Display Home Page

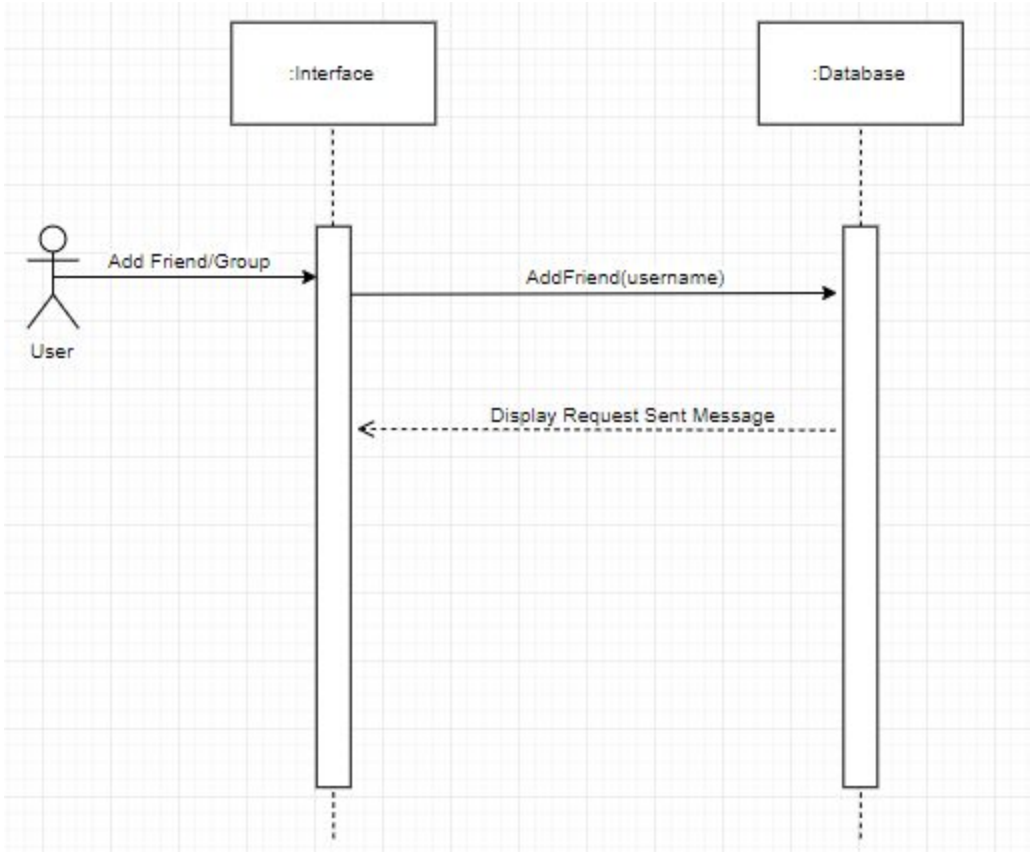
Login:


:Interface
:Database
User
Login
login(username, password)
validateLoginInfo(username, password)
Display Login Success Message
Display Home Page

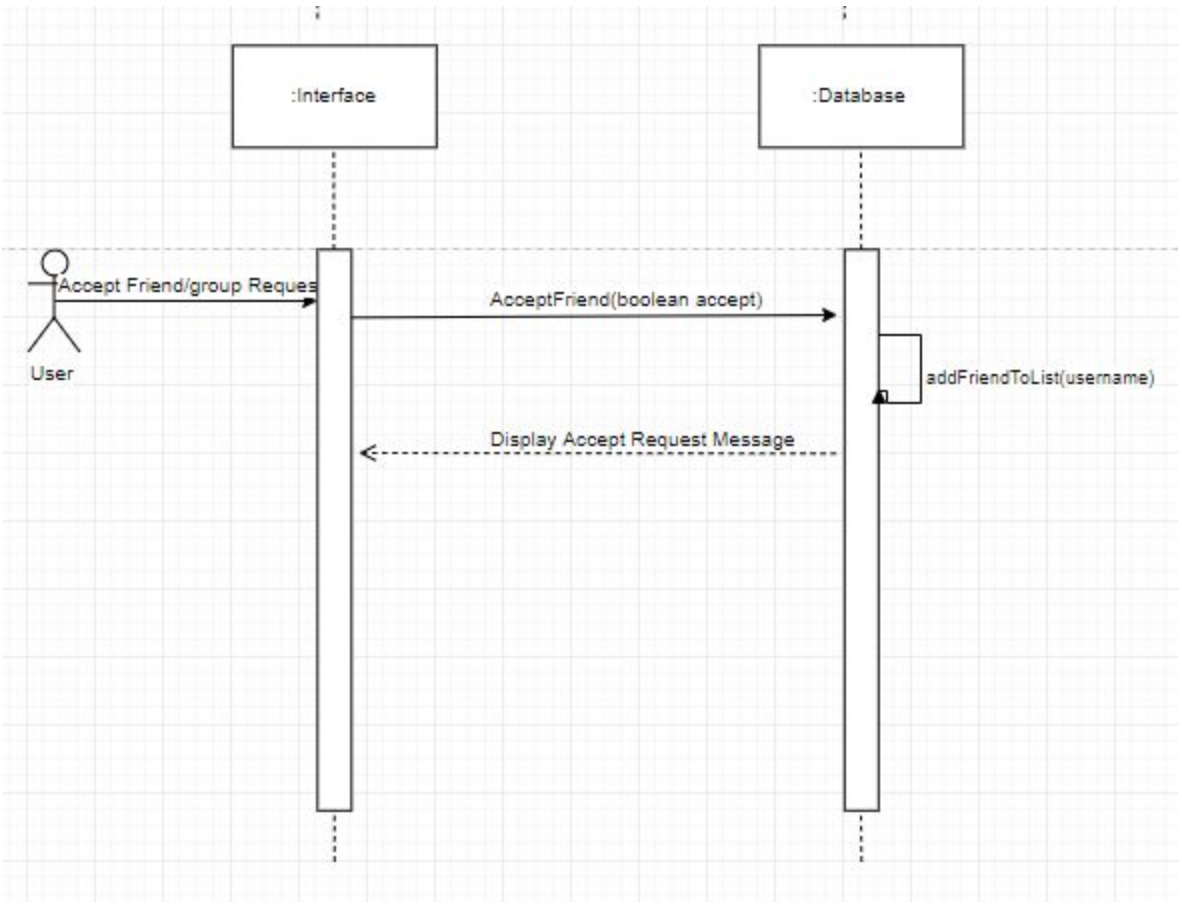
Create Post:



Add Friend / Group:



Accept Friend / Group Request:


:Interface
:Database
Accept Friend/group Reques
User
AcceptFriend(boolean accept)
addFriendToList(username)
Display Accept Request Message