

Challenge Your Friends

A social media friend challenging tool

Oscar Hedblad
Alex Berliner
Chris Kovaleski

Luis Duque
Jimmie Potts
Alan Birmaher



Please log in to Twitter

Email

Password

[Sign in or register](#)

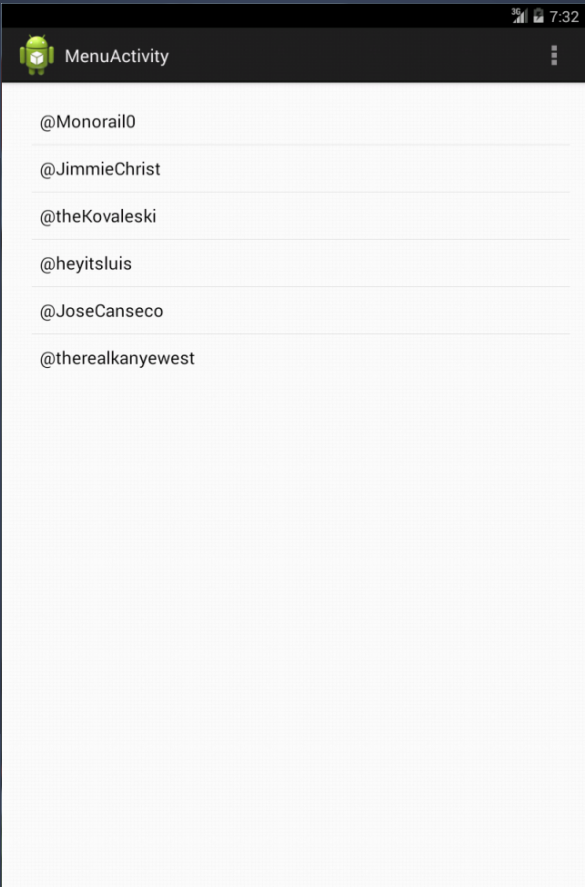


View challenges

Challenge friend

Complete challenge

Logout



@Monorail0

@JimmieChrist

@theKovaleski

@heyitsluis

@JoseCanseco

@therealkanyewest

Project Management Plan

Standards

For Coding:

- Java and Android coding standards
- Git for Version Control
 - Specifically the GitHub iteration for ease-of-use

For Documentation:

- IEEE to our abilities
- HTML format
 - Each page should reach every other page

Life Cycle

Development model: Agile

- Changes in specification
- Daily interactions with group
- Small changes
- Beneficial while learning
 - testing incrementally

Tools in the Environment

- Operating System: Our app will be supported on Android operating systems
- Programming Languages: We will be programming predominantly in Java
- IDE and Compiler: We will use the Eclipse IDE for Java Developers
- API: Twitter and Facebook

Maintaining Quality

Version Control:

- Changes made with Git version control

Risks:

- Obscene language and dangerous challenges

Concept of Operations

The Current System

- The Ice Bucket Challenge
- Challenge friends through social media, tag them, and set time limit for completion

The Proposed System: Needs

- Lack of unified platform for challenges
- Proposing application that lets user post and view challenges
- View most popular challenges
- Hashtagging to allow tracking of popularity
- Challenge through Twitter's tagging system

Users and Modes of Operation

- Only one mode of operation
- Pre-selected or customizable challenges
- Users upload to each other; no hosting req.
- User-wise, only handle uploading and accept/reject

Operational Scenarios: Post

- Users will enter and link their challenge video & give it a unique name
- Once named and linked, set time limit and number of people to challenge
- User then enters Twitter handles of their friends and tweets will automatically be generated
- Indecent Challenge(s)

Operational Scenarios: View

- View most popular challenges ranked through Twitter's trending algorithm
- View challenges posted by friends
- Rate challenges as they are viewing them

Operational Features

- Usable interface
- Users can challenge friends
- Post challenge videos
- Users can view the most popular challenges and rate them
- Simple rule system: \leq 140 characters

Would like to have:

- Back end for more robust data
- Nice, clean interface
- iOS support
- Ad support

Expected Impacts

- Access dedicated source of challenges posted by friends
- Because of IBC we strongly believe in the application
- Application with more challenges cannot go unnoticed
- Both on-campus as well as overall success

The Proposed System: Analysis

Improvements:

- Increased performance
- User interface
- Text-box with each challenge
- Profanity filter
- More challenges

The Proposed System: Analysis

Disadvantages:

- No verification of challenges being completed
- Multiple copies of similar/same challenges

Limitations:

- Required to be connected to Twitter
- Internet connection
- Time in which to develop the project

The Proposed System: Analysis

Risks:

- Obscene challenges (circumvented profanity filter)
- Dangerous challenges
- Application is dependent on Twitter API

Alternatives and Tradeoffs:

- iOS instead of Android
- Challenge your friends with pictures

Software Requirements Specification

System Requirements Specifications

Definitions, Acronyms, and Abbreviations:

- JRE: Java Runtime Environment
- JDK: Java Development Kit
- JVM: Java Virtual Machine

Assumptions:

- Users own or have access to an Android device.
- Users have a working internet connection from their mobile device.
- Users have a Twitter account.

Specific Requirements

3.1 Functional Requirements:

- Users shall open the Challenge your Friends app from their Android device
- User shall enter a mode in which an option to challenge friends or to accept
- In the case that the app crashes, the user shall relaunch the app and attempt to recover any data that was volatile at the time.

3.2 Interface Requirements:

- The app shall have a user interface.
- The user shall be able to create and edit challenges
- The user shall be able to send and receive challenges to others
- The user shall be able to alert others that they have completed a challenge
- The user shall be able to access the internet
- The user shall be able to upload media

Specific Requirements

3.3 Physical Environment Requirements:

- The application should be supported and run on Android mobile devices, with at least 1 GB of ram, a 1GHz processor, and internet access. The mobile device must have Internet Access to be functional.

3.4 Users and Human Factors Requirements:

- The user shall know how to use an android mobile device well enough to install apps
- The user shall be able to connect to the internet and has a twitter account.
- The administrators shall be able to manage a database. Documentation will be available for administrators in order to use/access server's resources.

3.5 Documentation Requirements:

- Instructions on how to launch the application will be presented with the application itself.
- Instructions on how to use the application will be provided with the application itself .
- Users who are reading the documentation are assumed to have a working understanding of social media and the platforms that are being utilized: Android and Twitter.

Specific Requirements

3.6 Data Requirements:

- Data about the user's existing challenges, both those sent to friends and those sent from friends will be retained.
- Private data regarding the user's twitter account information shall be stored.
- A system to rank and determine the popularity of challenges is essential for this application.

3.7 Resource Requirements:

- Administrators shall be trained to maintain and improve the software for the mobile application. The group will be available for email for basic issues with the application.
- Depending on how large the project becomes, we shall acquire skilled personnel to handle the server and for maintenance.
- We shall be using the agile development system to devise our schedules around. This system allows us to control development as we advance to maintain what we think should be the appropriate outcome.

Specific Requirements

3.7 Resource Requirements (Cont):

- We will be using the Android Development Tools for eclipse.
- We will be using the Twitter API to interface the android devices to the Twitter servers.
- Android devices are required for use in development.
- This project does not have major operating costs because it is being hosted on Twitter, a free platform.

3.8 Security Requirements:

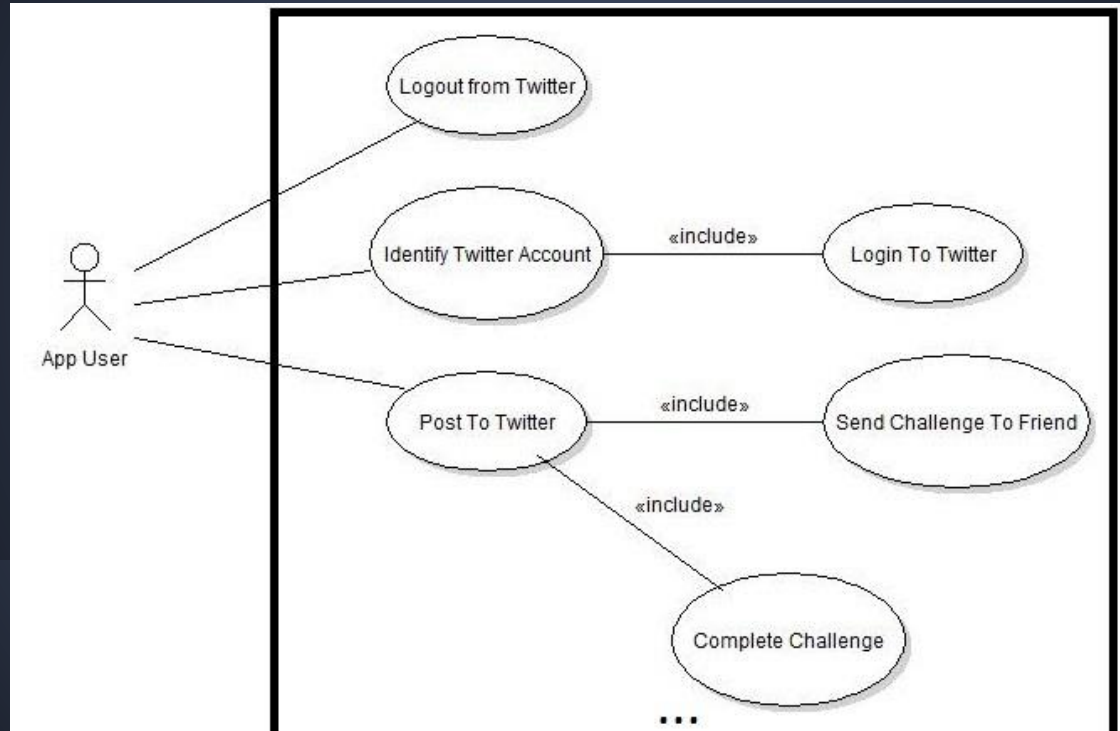
- Before being able to access the content of the application, users shall be able to log in using an existing Twitter account. This will be prompted for upon the application launch.
- User data shall be isolated from others.
- We shall determine a reasonable amount of time to backup the system based on the amount of server capacity available.

Specific Requirements

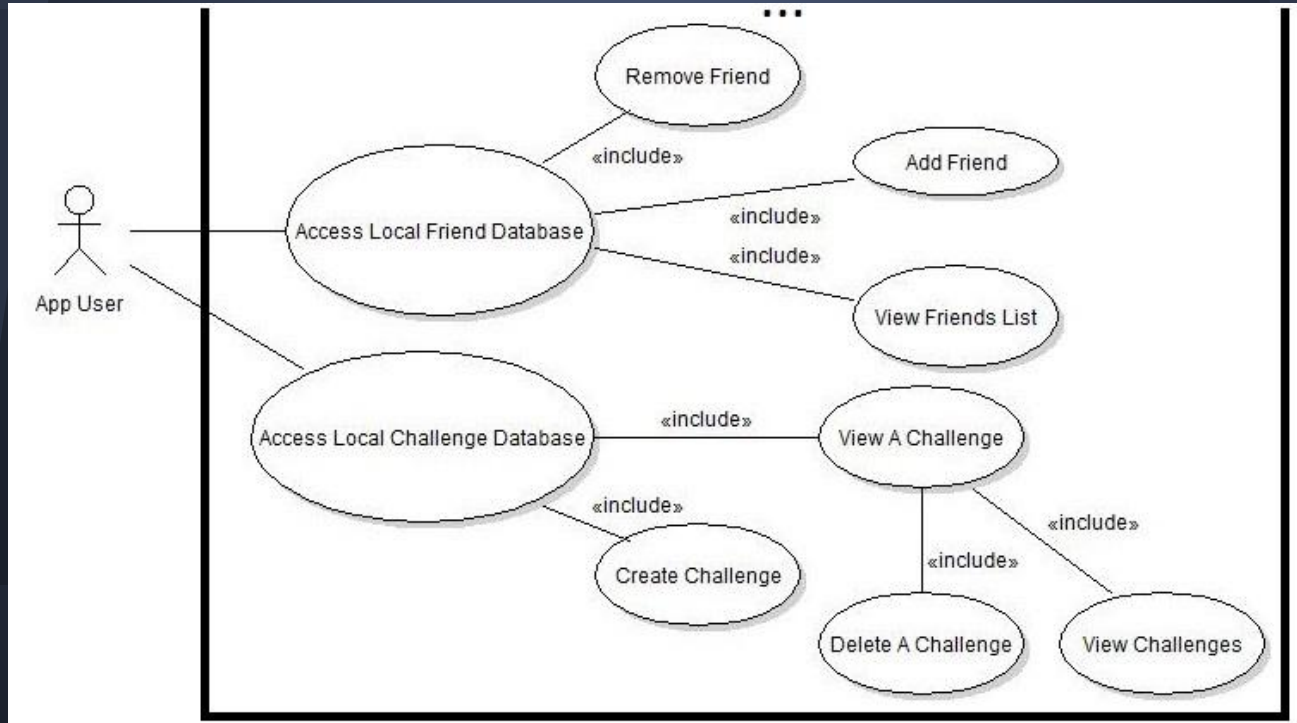
3.9 Quality Assurance Requirements:

- The system shall be available at all times. This is easily accomplished because the service will be hosted entirely on Twitter.
- Faults in the system shall be in the realm of data tracking and whether twitter's algorithm changes to edit their trends.
- We shall be able to fix faults in the system by pushing an updated version of our app to the app store
- Resource usage and response times only matter on the android device. Fast response times are easily accomplished on the android device based on the frequency by which the app polls twitter.

Use Case: Twitter Interface



Use Case: Database Interface



System Requirements Specifications

Event Table Overview

- App starts: User logs into Twitter to access their account
- User creates challenge: Challenge is stored to phone
- User sends challenge: App delivers challenge to friend
- User receives challenge: App alerts user of challenge
- User accepts challenge: App updates challenge status
- User completes challenge: App asks to share challenge
- User shares challenge: App uploads media of challenge completion

Test Plan

Objectives for Software Test Activity

Uncover and address issues by way of internal testing, followed by external beta testing in the following areas:

- Software Bugs
- Security
- Easy of Use
- Usefulness/ Practicality

Test Environment

- Multiple Android Devices
- Test Stages
 - Alpha: Internal/ Developer Testing
 - Beta: External testing along with continued internal testing

Stopping Criteria

- Minor Bugs
- Major Bugs
- Improvements
- Stop Development

Individual Test Cases

Initial test cases will be various combinations of pair from the following 2 groups:

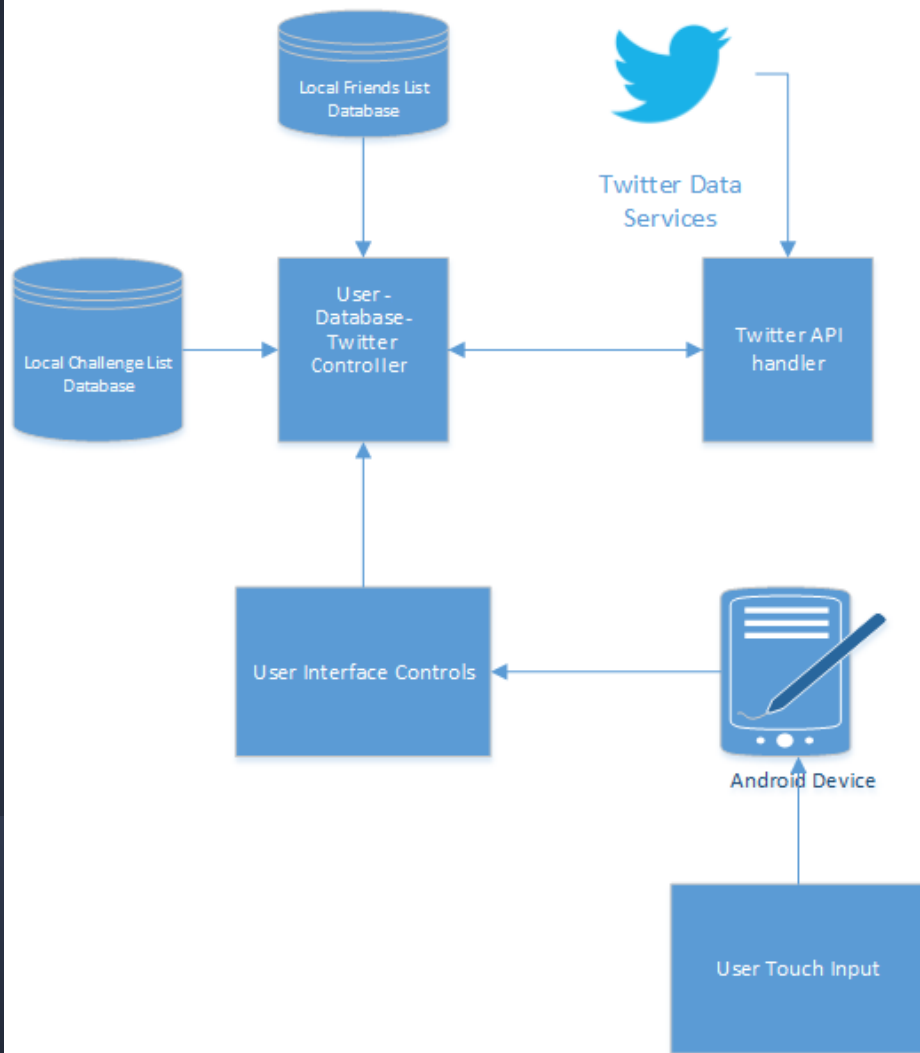
What:

- Initiate challenge to individual
- Initiate challenge to group
- Respond to challenge and challenge one user
- Respond to challenge and challenge multiple users

Via:

- Twitter
- Facebook
- Google+
- Instagram

High Level Design



Design Issues

- **Reusability**
 - Object-oriented design
 - Using open source libraries such as Twitter4J
- **Maintainability**
 - Well written documentation
 - Proper commenting of code
- **Testability**
 - Test throughout development
 - Object oriented testing

Design Issues

- **Performance**
 - Reliant on Twitter
- **Portability**
 - Main goal: Android platforms
- **Safety**
 - Ensure that illegal/dangerous challenges are discouraged

Technical Difficulties

- Twitter API
 - Open source Twitter libraries
- Android SDK
 - Online tutorials and previous experience
- Database implementation
 - Keep it small and simple

Design Trade-Offs

- YouTube for video hosting
- Twitter hashtags

Detailed Design

Issue Relevance

- Testing-related issues will be evaluated while in the testing-stages of the process of development.
- All project interfaces should be built in an easy manner so that testing can be done smoothly.
- Safety issues are next to minimal.
- Portability to other devices. A goal is to get our application to run on iOS, BlackBerry, and Windows.

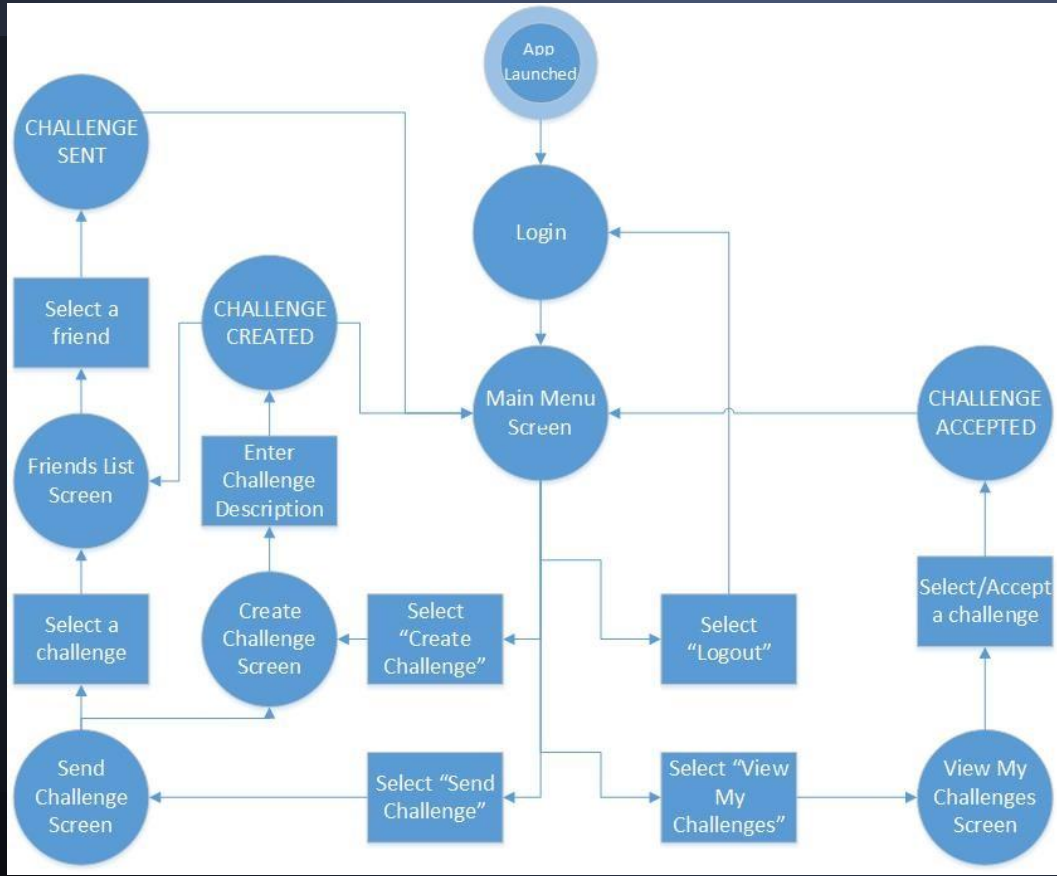
Expected Technical Difficulties

- Getting the API to work properly might take more effort than planned. Members have different technical skills.
- Having the users upload videos directly to YouTube brings in YouTube-related risks, such as something being wrong with their server, etc.

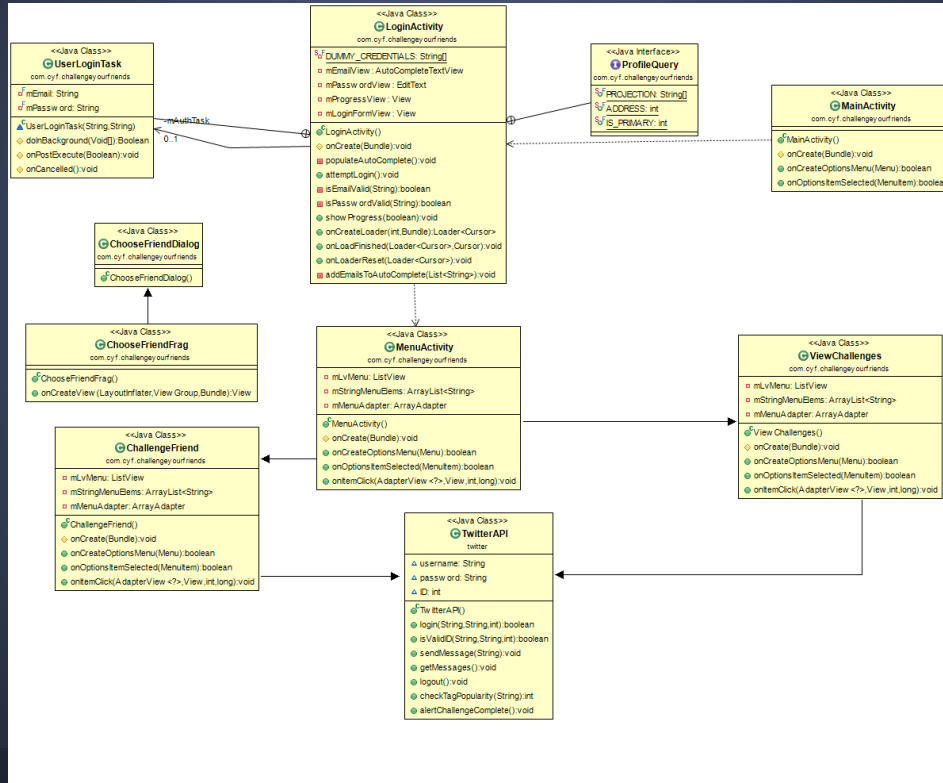
Design Decisions and Associated Risks

- Java
 - Supported by Android
 - Risk: Improper coding syntax
- Android
 - Most accessible
 - Many android devices
- Twitter
 - One of the largest social media platforms
 - Most appropriate for our app
 - Risk: App heavily relies on twitter

Application State Diagram



Application UML Diagram



Send challenge sequence

