TAPWATER

SOFTWARE SYSTEM DESIGN

TEAM MEMBERS:

SHIH CHAO
DAVID FONTANA
JONATHAN HOOPER
CODY ROGERS

I. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of the TapWater project in order to foster further understanding of project elements amongst programmers.

1.2 Overview

The following document is organized into several sections:

II. Main system architecture description

III. Sub-systems architecture description

IV. Rationale for each architectural design choice

V. Development view diagram

VI. Physical view diagram

VII. Data view diagram

VIII. Work-Assignment diagram

IX. Element catalog

X. User Interface

II. MAIN SYSTEM ARCHITECTURE

TapWater's main system structures run on a "client-server" model architecture. The system's clients consist of the iOS and Android applications, both of which communicate through a Heroku server. The Heroku server provides data access across both client systems and maintains data integrity.

III. SUB-SYSTEMS ARCHITECTURE

TapWater's iOS and Android application sub-systems run on an object-oriented architecture. All data representation objects are completely self-contained. See the data view (VII) and element catalog (IX) for additional information regarding the individual subsystems.

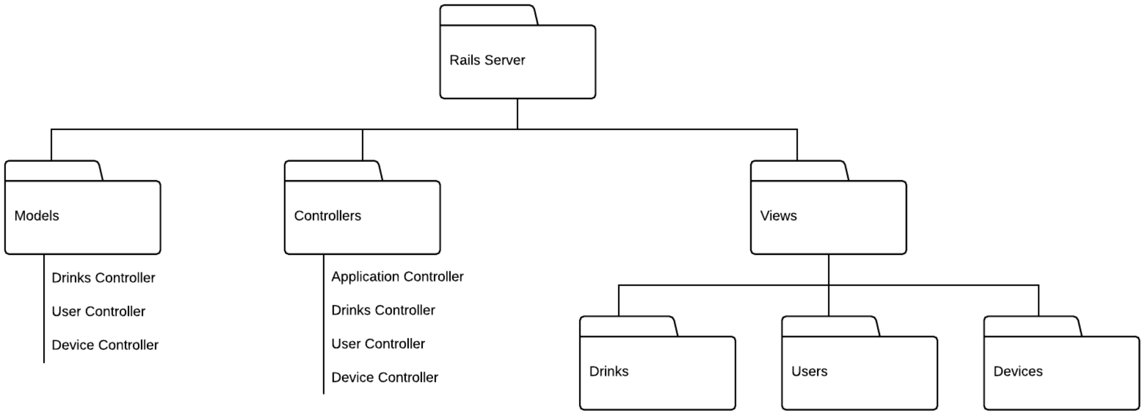
IV. RATIONALE

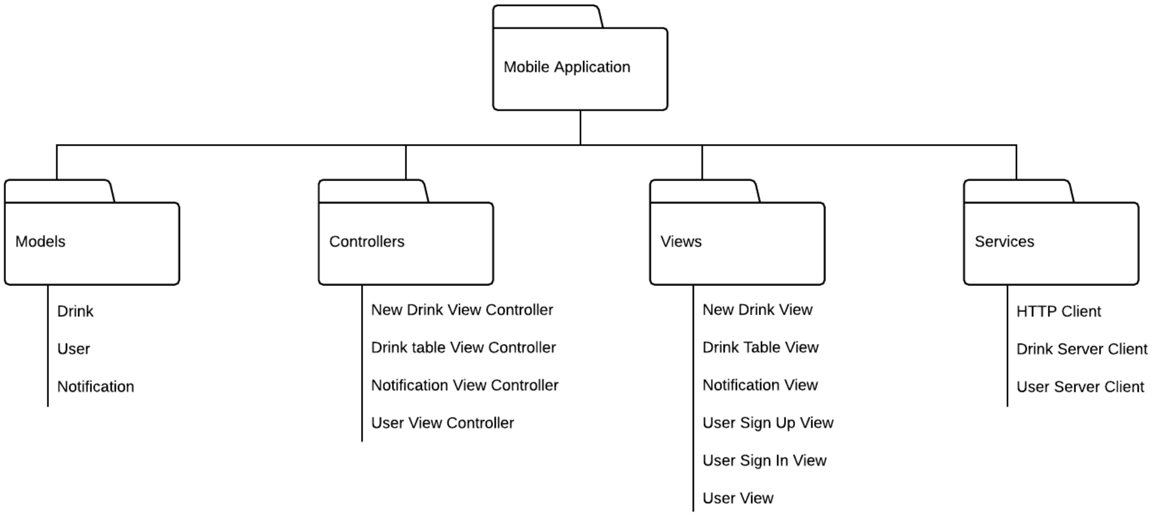
The Client-Server architecture style was selected as the main system architecture because TapWater's design requires the ability to send, store, and receive data on drinks to and from a client device. This made the Client-Server architecture ideal to the situation.

The Object-Oriented architecture style was chosen for the subsystem because of the need for a simplistic and self-contained data format to facilitate cross-platform synchronization and to keep future development implementations easier.

V. DEVELOPMENTAL VIEW

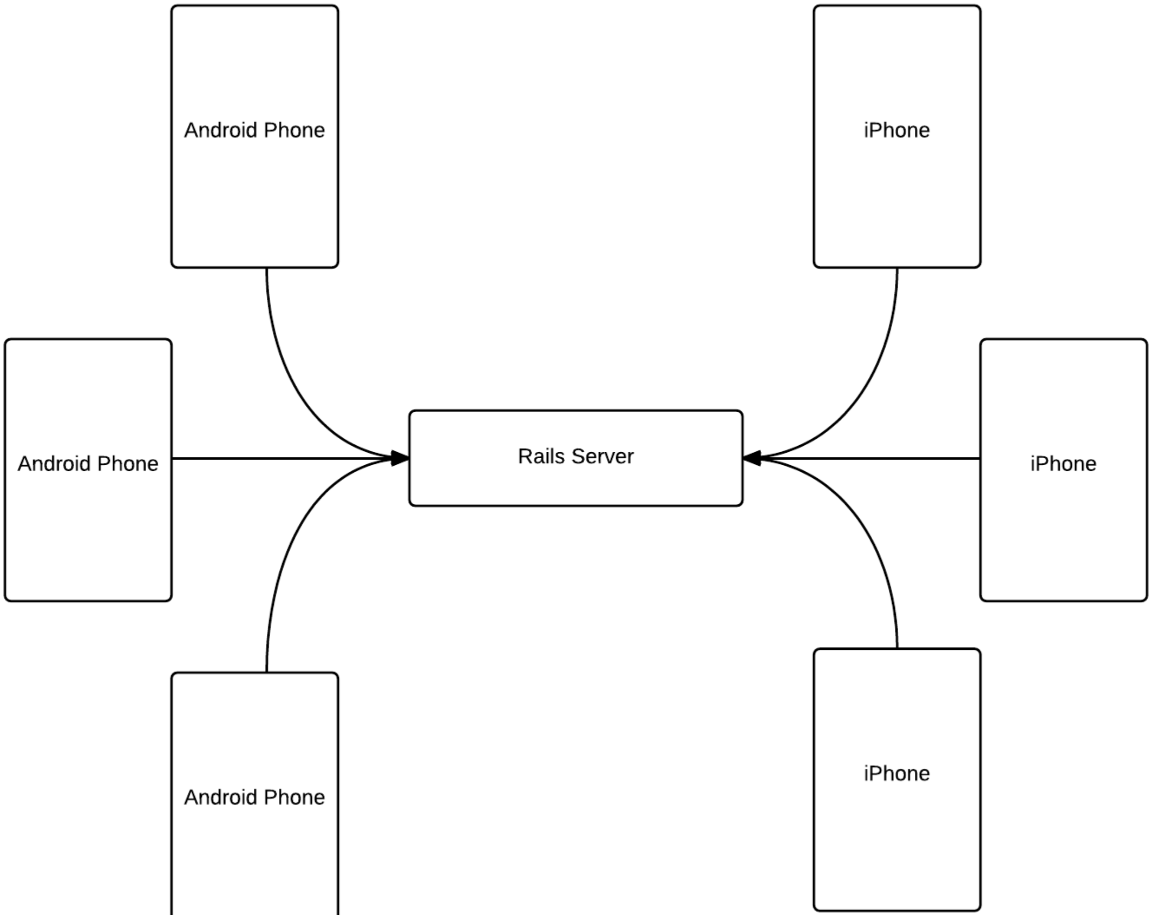
The following diagrams break down the organization of both the Server and Client systems and their components.





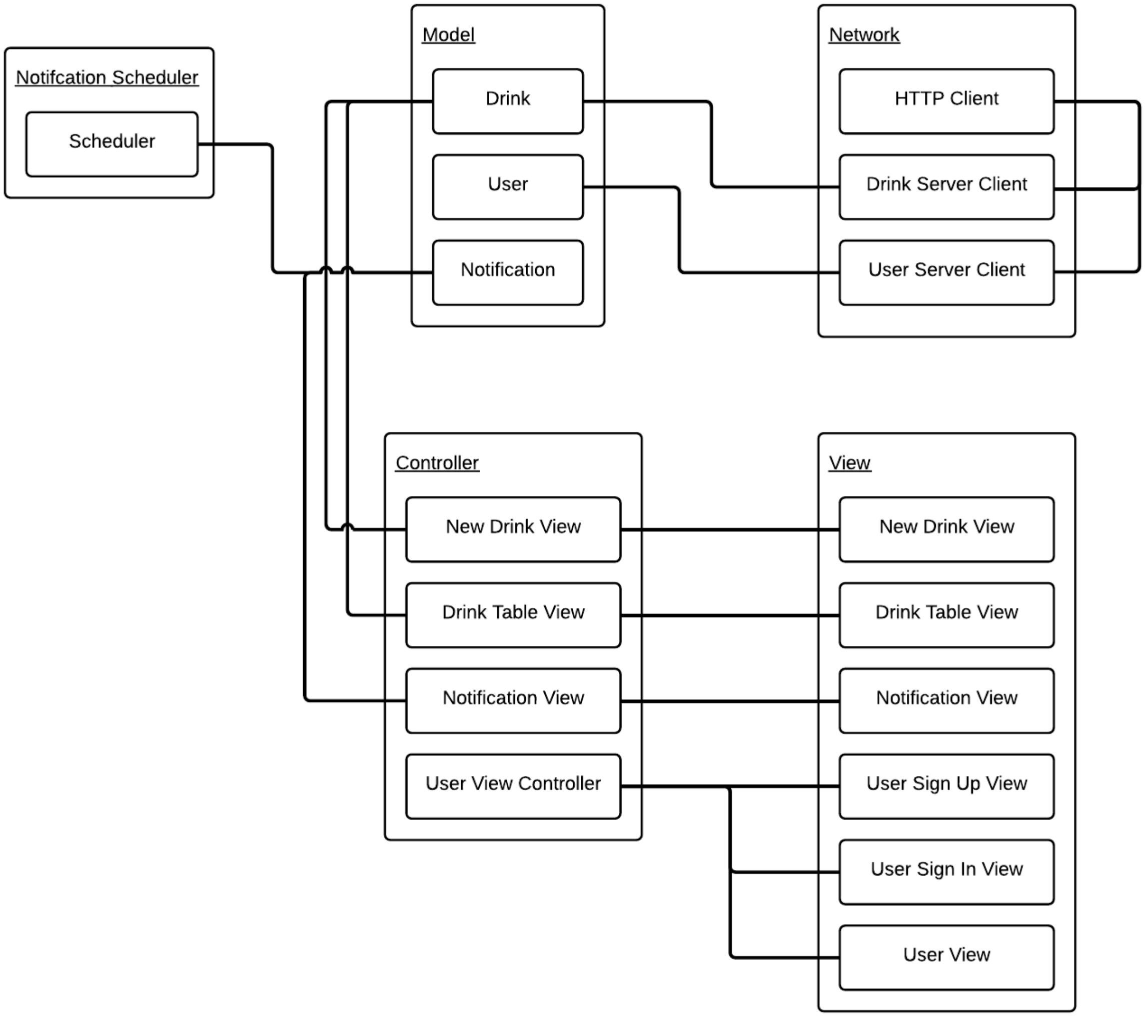
VI. PHYSICAL VIEW

The following diagram depicts the overall physical structure of the intended system. Here, the “Rails Server” (Heroku) is the Server and each of the iPhone and Android devices are the Clients.



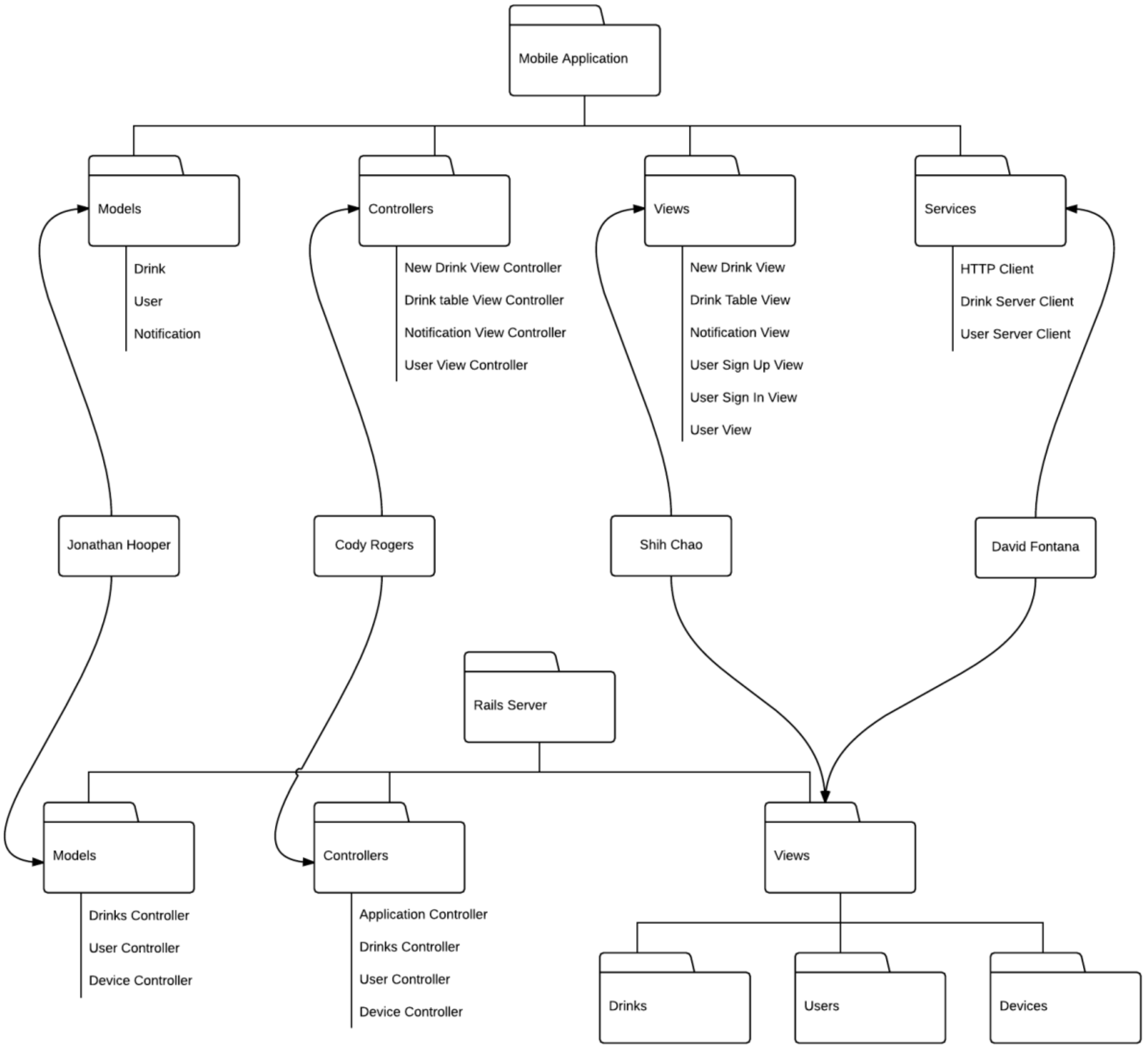
VII. DATA (LOGICAL) VIEW

The following diagram showcases the interaction between and structure of systems and subsystems.



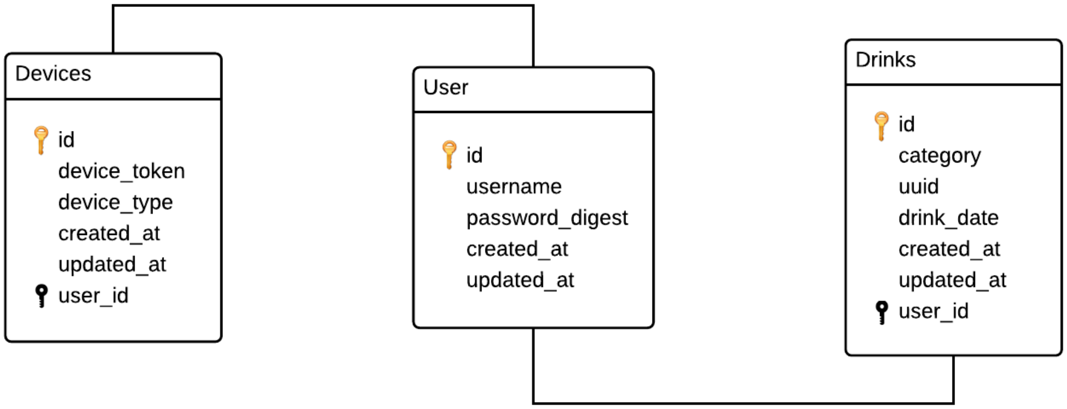
VIII. WORK-ASSIGNMENT VIEW

This diagram illustrates the different work-assignments amongst group members and how each assignment integrates into the project as a whole by utilizing the developmental view.



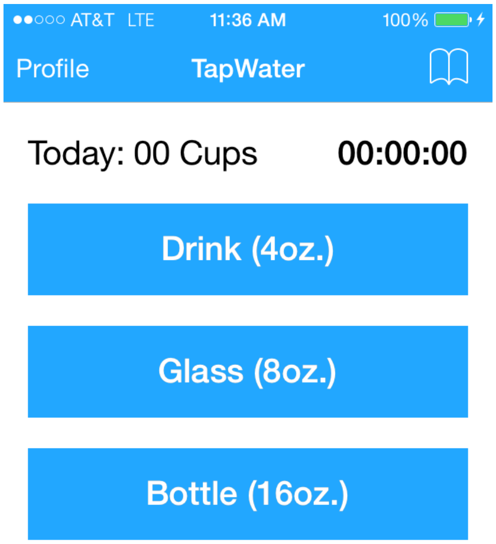
IX. ELEMENT CATALOG

This final diagram shows the internal components of each data object representation (namely: Devices, Users, and Drinks).



X. USER INTERFACES

As seen here, the user interfaces (encountered only on client devices) are kept deliberately simple and minimalistic in order to facilitate ease-of-use amongst the entire user base. That is, one of the TapWater project's goals is to be incredibly easy to dive into as a new user with very little learning curve.
The interface consists primarily of two screens show below. The first, main screen is where users will add drinks to their daily count. The second screen is a drink history which showcases when each drink was added to the user's list.



●●○○○ AT&T LTE 11:36 AM 100%

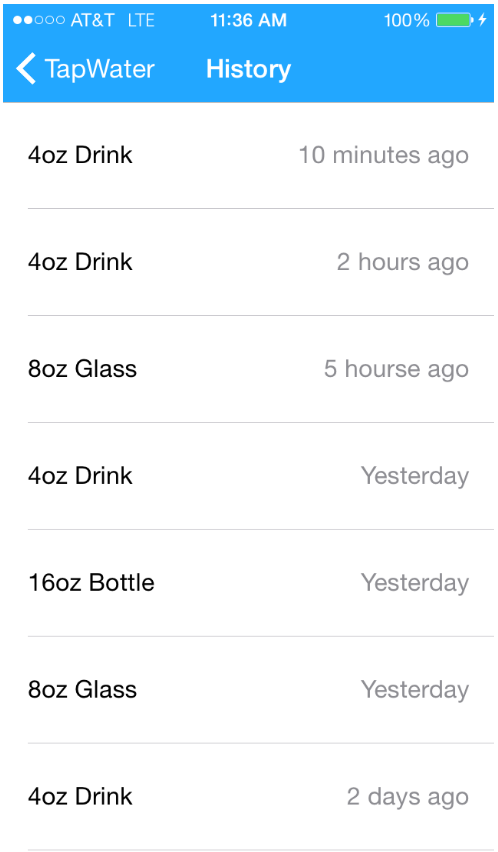
Profile **TapWater**

Today: 00 Cups **00:00:00**

Drink (4oz.)

Glass (8oz.)

Bottle (16oz.)



●●○○○ AT&T LTE 11:36 AM 100%

‹ TapWater **History**

4oz Drink	10 minutes ago
4oz Drink	2 hours ago
8oz Glass	5 hourse ago
4oz Drink	Yesterday
16oz Bottle	Yesterday
8oz Glass	Yesterday
4oz Drink	2 days ago