
Team 3

Brewers Buddy Website

Software Architecture Document

Geoffrey Blogref
Gregg Ideus
Jonathan Parise
John Pistorius
Steve Platz

Version 1.0

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

Revision History

Date	Version	Description	Author
8/12/2013	1.0	Initial Version	Team 3

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

Table of Contents

1. INTRODUCTION	4
1.1 PURPOSE.....	4
1.2 SCOPE	4
1.3 OVERVIEW.....	4
2. ARCHITECTURAL REPRESENTATION.....	5
2.1 BUSINESS GOALS.....	5
2.2 SYSTEM CONTEXT	6
2.3 SYSTEM USE CASE	7
3. ARCHITECTURAL GOALS AND RESPONSIBILITIES.....	7
3.1 FUNCTIONAL RESPONSIBILITIES.....	7
3.2 <i>CONSTRAINTS</i>	8
3.3 ARCHITECTURAL DRIVERS	8
4. LOGICAL VIEW	9
4.1 OVERVIEW.....	9
4.2 ARCHITECTURALLY SIGNIFICANT DESIGN PACKAGES	11
4.2.1 <i>Application Layer</i>	11
4.2.2 <i>Data Layer</i>	12
4.2.3 <i>Presentation Layer</i>	13
4.2.4 <i>Service Layer</i>	14
5. DESIGN	15
5.1 CONTRACT	15
5.2 INTERACTION DIAGRAM	16
5.3 CLASS DIAGRAM	16
5.4 STATE CHART	17
6. RATIONAL.....	18

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the Brewers Buddy system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

This document applies to the overall design of the system. It contains information relating to the architectural design of the software, the Structure of the Database, and of the physical servers hosting the site.

1.3 Overview

The Brewers Buddy Website was designed to help home brewers keep track of their home brews from start to finish. Home brewers need a place to keep their recipes, batches, and inventory numbers safe. They also would like to experiment from other batches created by other users as well as share their own. Brewers also wanted a unified software solution that could unite all home brewers in the street, neighborhood, city, or further. Finally, home brewers wanted a place where they could keep all their documents and not worry about losing them or destroying them.

The system described in the following sections seeks to meet, and exceed, the expectations that home brewers are wanting, and provide a solution, to those home brewers, that surpass their previous way of keeping track of their home brews.

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

2. Architectural Representation

This system was designed using a standard three tiered architecture, with a Presentation Layer, Business Layer, Data Layer, and the Service Layer.

- **The Presentation Layer** -- contains all of the visible web pages and handles all input from and output to the user.
- **The Business Layer** -- handles all of the Business logic and provides an abstraction to the database.
- **The Data Layer** -- consists of the Database and stored procedures contained within, and this provides the persistence required for the system.
- **The Service Layer** -- A Service Layer defines an application's set of available operations from other interfacing client layers

2.1 Business Goals

The business goals for the Brewers Buddy Website are as follows.

- Build and maintain a site that provides beer and wine brewers with an online tracking system that is accessible from any internet-connected device.
 - By keeping in mind the types of devices that can be used to access the internet during all levels of the design process; we hope to build a system that home brewers will feel confident using from any of their devices to access their information.
- Provide the brewer's with an interface to register with the system.
 - Allowing the brewers to easily register for the site will be a good start. Ease of use and limited complications is a good way to keep the brewers happy with the site
- Provide a way to record and display a brewer's home recipes.
 - An easy to read display of each of the brewer's home recipes will improve the brewer's experience with the site
- Provide a way to record, display, and share a brewer's batch information.
 - An easy to read display of each of the brewer's batch information will improve the brewer's experience with the site.
- Provide a way to record and display notes, measurements, comments, ingredients, and ratings for a brewer's saved batches.
 - An easy to read display of each of the display notes, measurements, comments, ingredients, and ratings for a brewer's saved batches will improve the brewer's experience with the site.
- Provide a way to record and display a brewer's inventory of completed home brews.
 - An easy to read display of each of the brewer's inventory will improve the brewer's experience with the site.

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

2.2 System Context

The Brewers Buddy System's overall goal is to centralize a place for home brewer's to keep track of their brew data and make it easily accessible for them. The brewer's data is stored in a database and is accessible from many platforms over the internet. Using various devices (computers, tablets, and smartphones) users can view their recipes, batches, and inventory.

System availability is a high priority; brewer's information always needs to be accessible. To achieve this, Azure site hosting is used. This will make the site accessible at any time and will damper the fear of the system being overburdened. The overall context of the Brewers Buddy system is shown below.

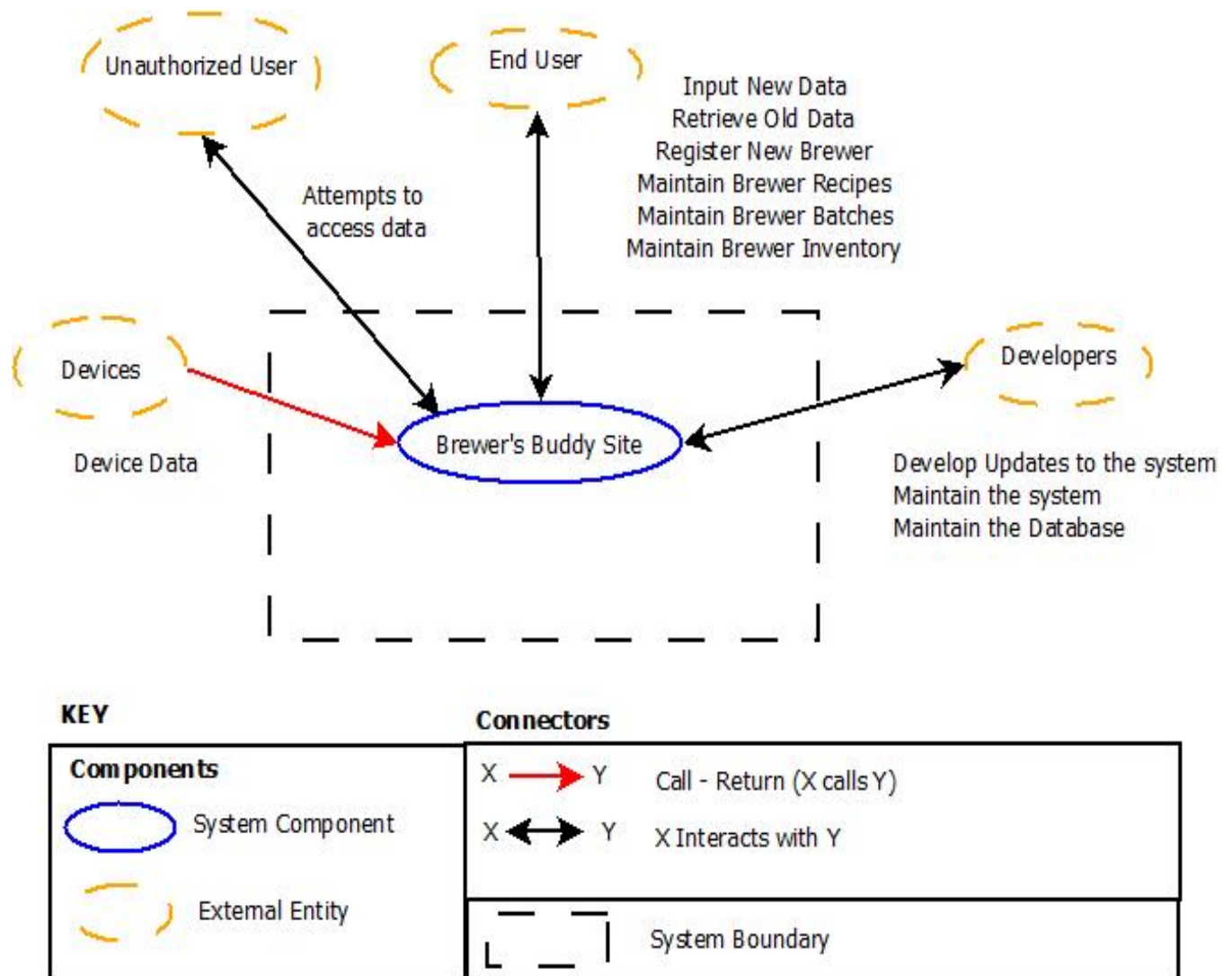


Figure 1: Brewers Buddy Context

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

2.3 System Use Case

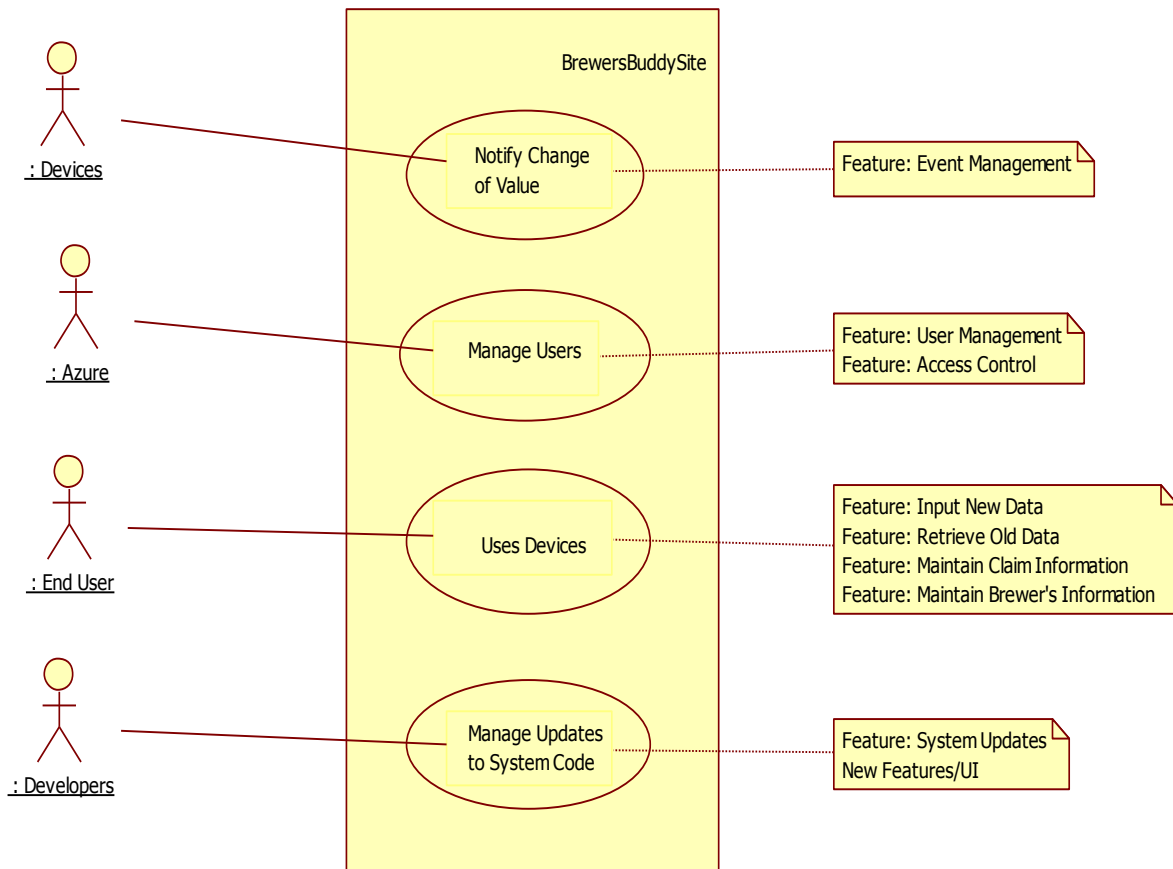


Figure 2: Brewers Buddy System Use Case

3. Architectural Goals and Responsibilities

The goal of this project was to create a website capable of allowing home brewers a digital place to create, store, and view their concoctions. The system is to allow a user to create a profile, allow them to add recipes, allow them to create batches, be able to add notes to each batch, and finally keep tabs on their inventory. The constraints we had making this was the limitation of the MVC framework and the learning curve for it.

3.1 Functional Responsibilities

- Create brewers accounts
- Authenticate and authorize brewers
- Edit brewers accounts
- Remove or suspend brewers accounts

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

- Encrypt all sensitive or private user information
- Allow search for other brewers
- Allow adding of other brewers as friends
- Allow entry of brewers recipe data
- Allow entry of brewers batch data
- Allow entry of brewers ingredients data
- Allow entry of brewers inventory data
- Allow entry of brewers notes for batches

3.2 Constraints

Category	Factor	Description
Technology	Scalability and Modifiability	System must be scalable to handle various platforms and easily modifiable to accommodate newly invented platforms
Technology	Responsiveness	System must maintain 99.9% uptime
Technology	Devices	System must support all approved devices
Technology	Responsiveness and Performance	System must handle a large number of simultaneous users
Technology	Devices	System must support all approved devices for users
Regulation	Compliance	System must comply with government issued regulations
Regulation	Compliance	System must comply with requirements set forth in the Computer Matching and Privacy Protection Act of 1988
Product	Performance and Capacity	System must be able to handle a large number of devices without decreasing its performance
Product	Usability	System should be accessible to all users

3.3 Architectural Drivers

#	Architectural Driver	Priority
1	Responsiveness of system	(H, H)
2	Fault tolerance under load	(M, H)
3	Allow adding of new brewers	(H, H)
4	Allow adding	(H, H)
5	Support for cross-platform compatibility	(H, H)
6	Support for real-time brewer collaboration	(H, H)

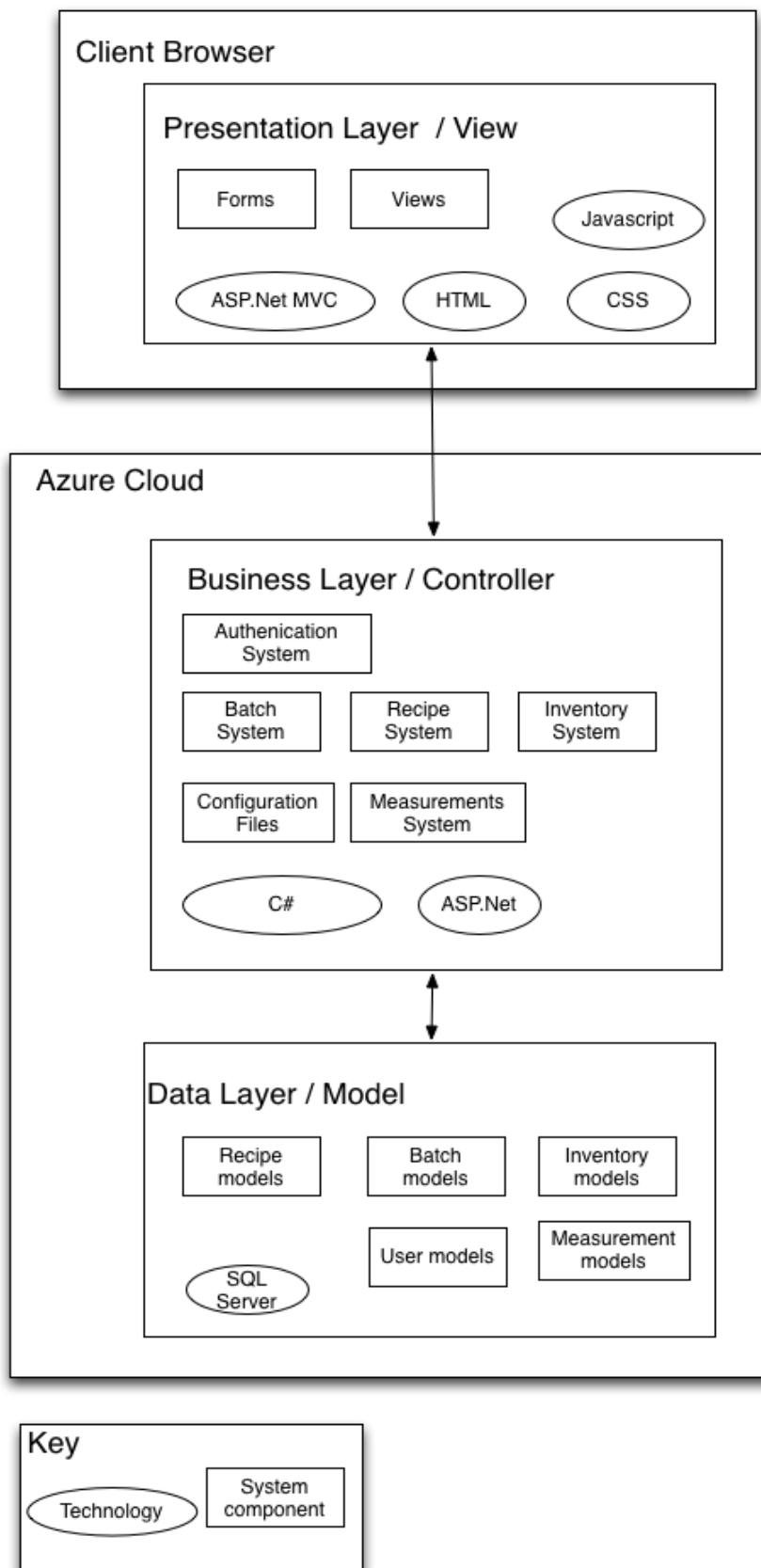
BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

4. Logical View

4.1 Overview

The Logical View shows a quick overview of all of the basic subsystems in the System and gives a basic overview of the System as a whole.

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013



BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

Figure 3: High Level Context

This Diagram shows the various high level packages that the system is broken down into. The packages can be seen here separated into the various layers.

4.2 Architecturally Significant Design Packages

4.2.1 Application Layer

The Application Layer provides the business logic and connects the Presentation Layer to the Database. The Application Layer is contained within the Brewers Buddy models. All communication with the Presentation layer is done through Services.

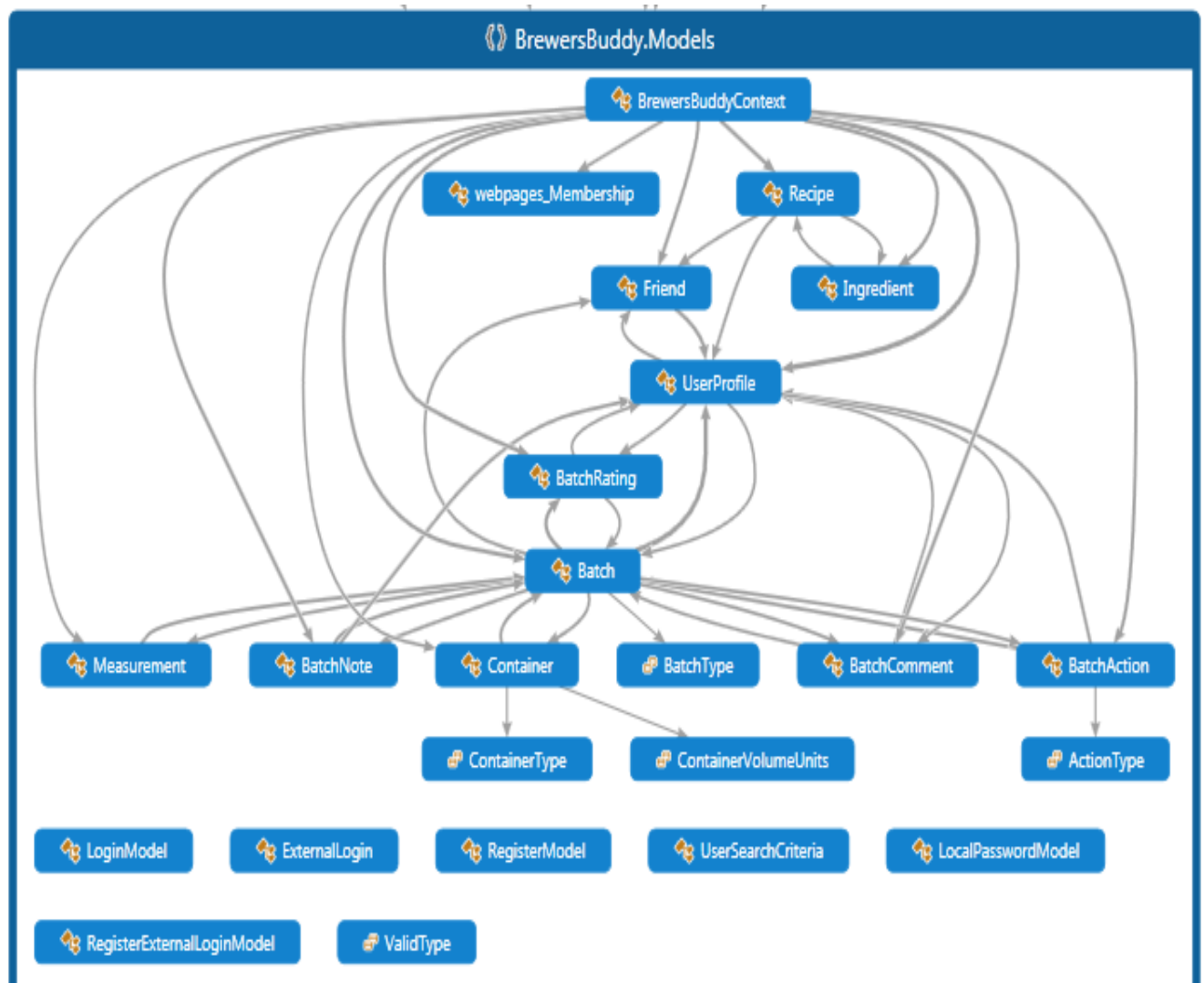


Figure 4: Model Layer of the System

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

4.2.2 *Data Layer*

The Data Layer consists of the Microsoft SQL Express Database and the stored procedures contained within. The Data Layer provides persistence for the system and all communication is done by the MVC framework. See Data View for data-base structure.

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

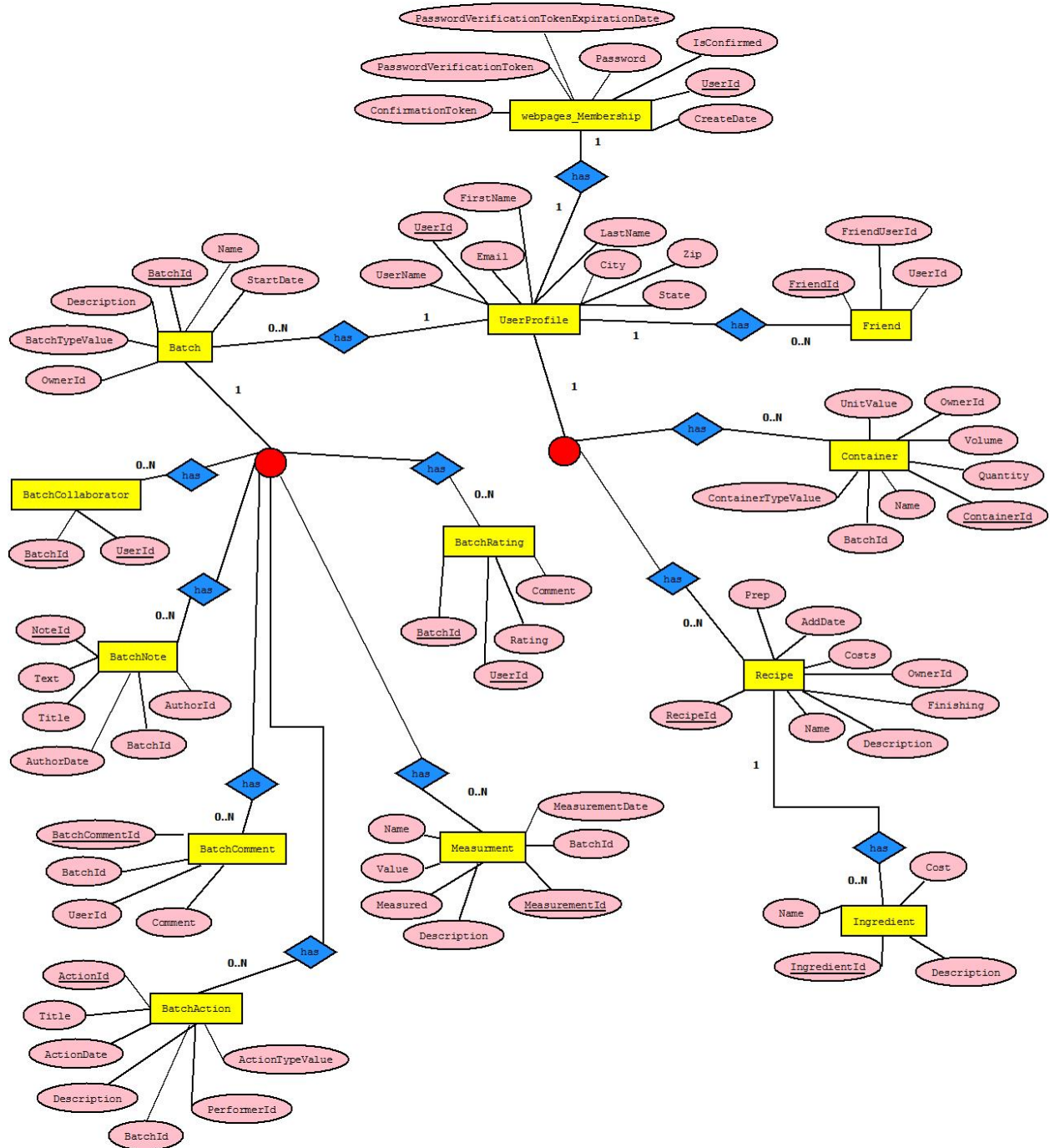


Figure 5: Data Access Layer for Brewers Buddy Website

4.2.3 Presentation Layer

The Presentation Layer consists of the controllers in the MVC framework and those that have been added. They display the data to the end user and allow them to manipulate the data and save it. The graphical representation of the website consists of many controllers.

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

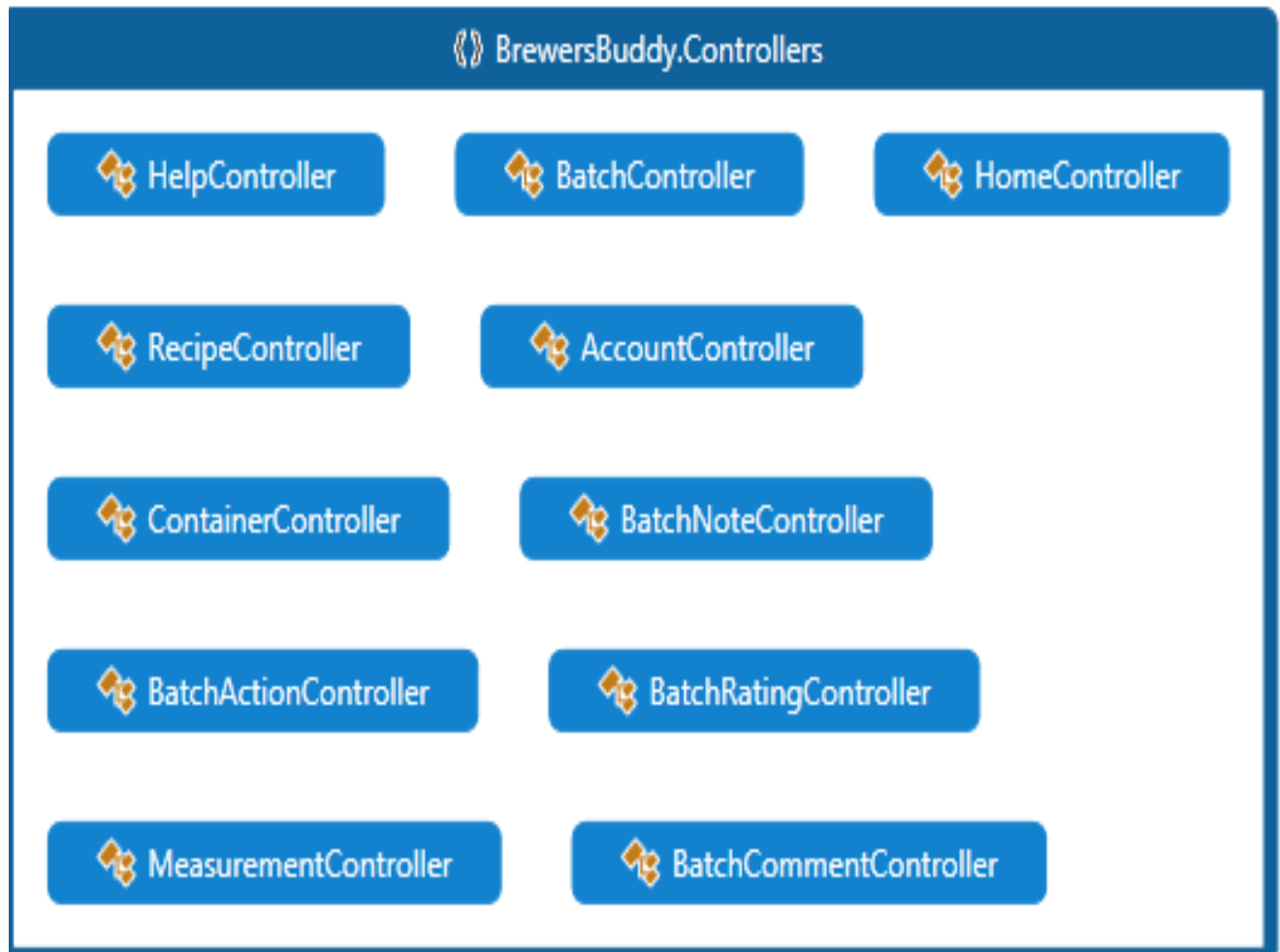


Figure 6: Controllers in the Brewers Buddy Website

4.2.4 Service Layer

The Service Layer consists of the services used for the inbound calls from the presentation layer. This layer consists of interfaces acting as a façade for exposing the business logic to the brewers using the site.

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

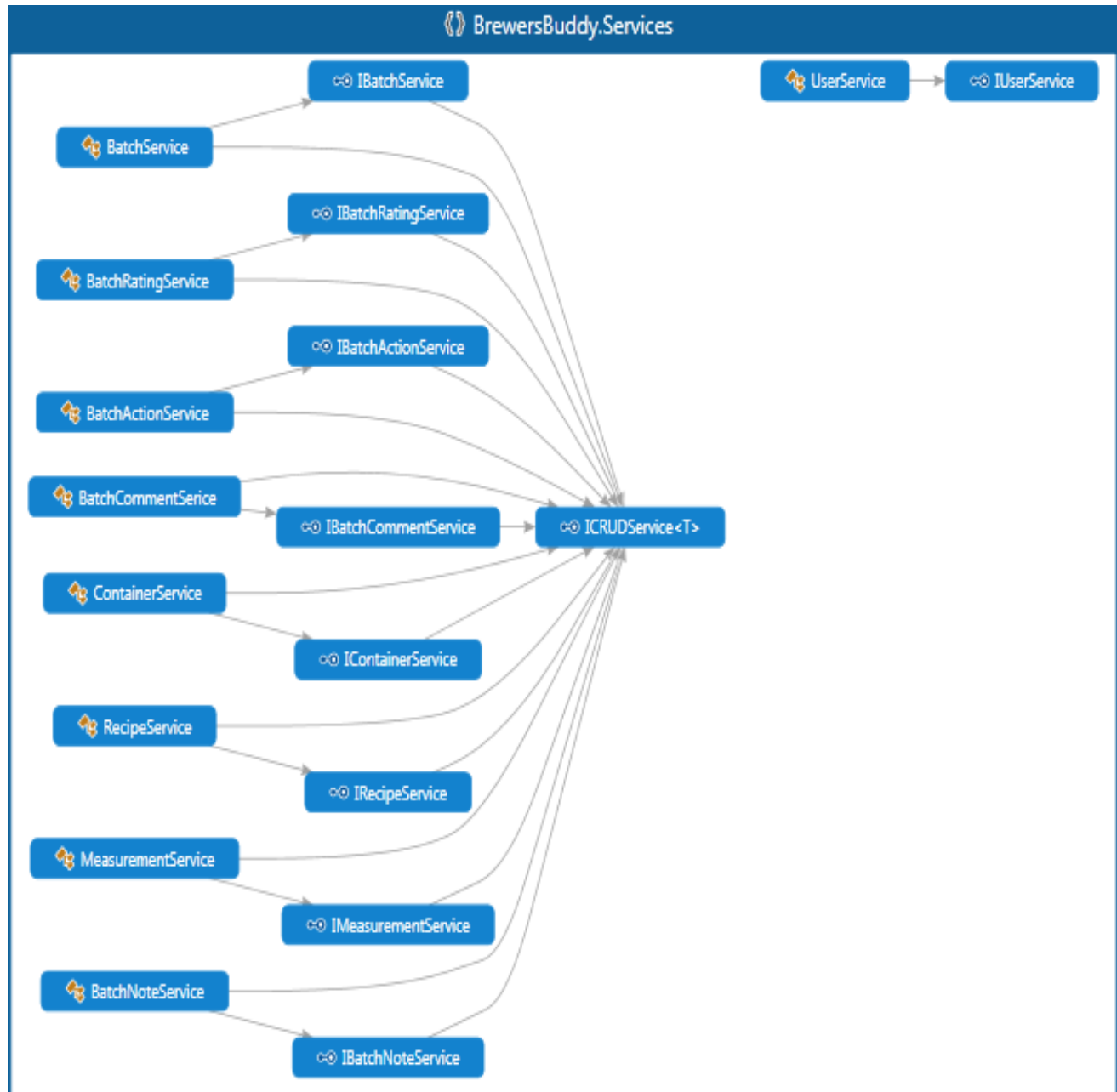


Figure 7: Services Layer for Brewers Buddy Website

5. Design

5.1 Contract

Name: createNewBrewerProfile()

Responsibilities: Generates a new brewer within the System.

Type: System

Cross References: Use Cases: Create New Brewer

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

Notes: none

Exceptions: none

Output: none

Pre-Conditions:

- A new brewer wants to join the site

Post-Conditions:

- A new brewer account is made
- New brewer is authenticated and logged in
- New brewer is now able to create recipes
- New Brewer is now able to create batches
- New Brewer is now able to search for other brewers
- New brewer is now able to add to inventory

5.2 Interaction Diagram

The following interaction diagram displays the activities performed during the createNewBrewerProfile contract

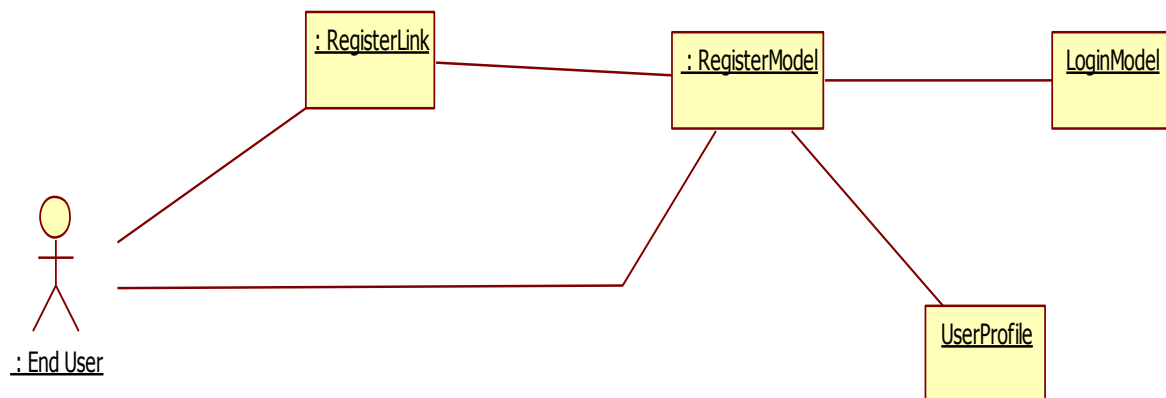


Figure 8: Brewers Buddy Interaction Diagram

5.3 Class Diagram

The following static class diagram is the initial system design based on the createNewBrewerProfile contract

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

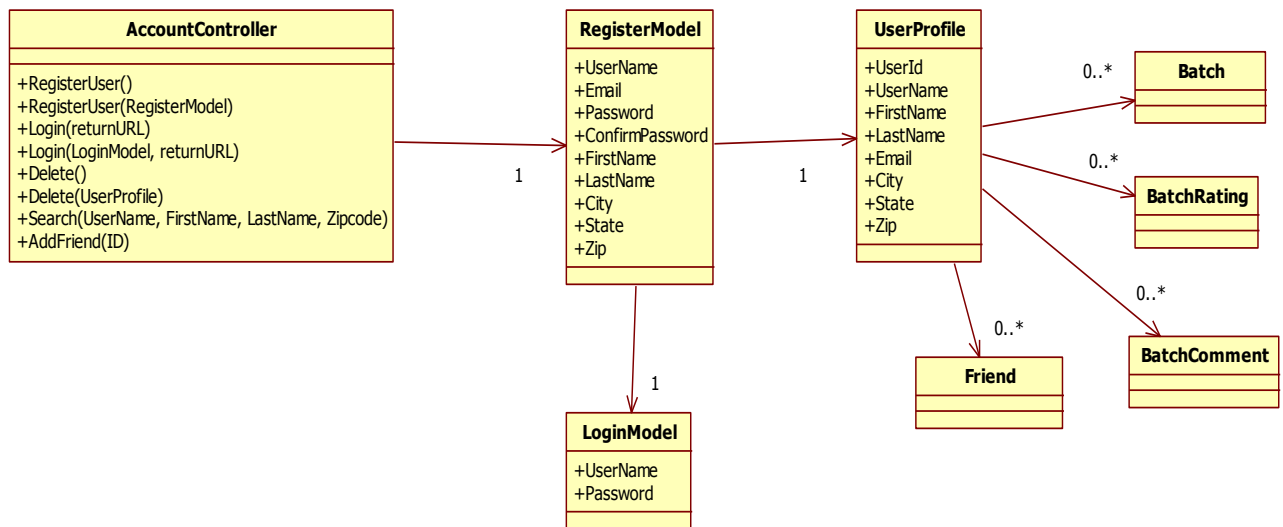


Figure 9: Register User Class Diagram

5.4 State Chart

The following state chart shows the various stages the system will be in by the AccountController class

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

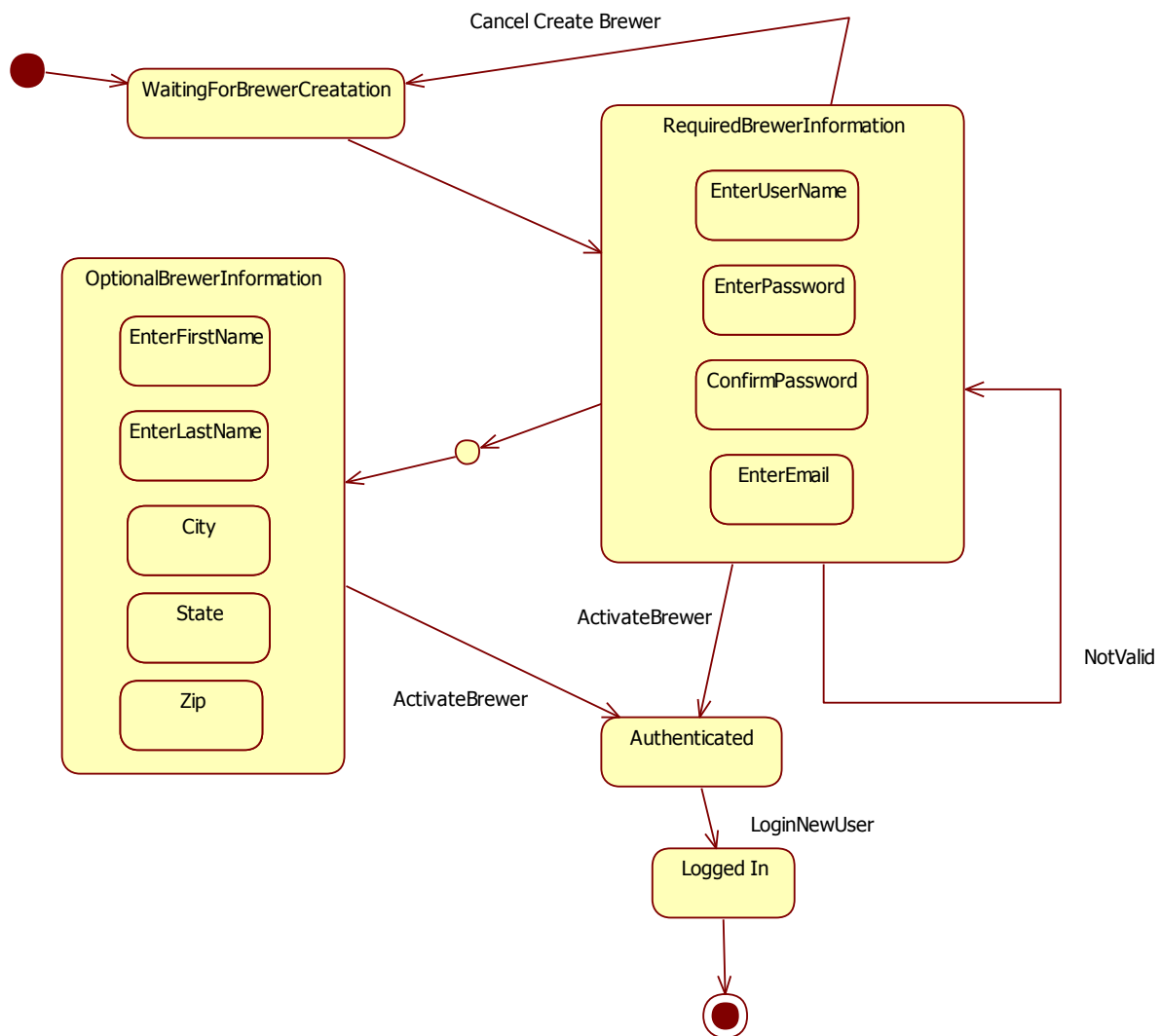


Figure 10: Register new brewer

6. Rational

As the Brewers Buddy Website becomes more popular and offers its services to a broader community spanning multiple time zones, it is necessary to have a centralized system that can sustain growth and support new market areas. Due to the growing number of home brewers a centralized access point to brewer data is important to provide prompt services. In addition to that there is a need to increase efficiency and productivity of the website.

Key Features:

BrewersBuddy	Version: 1.0
Software Architecture Document	Date: 08/12/2013

- Centralized information: Provide brewer's anytime access to personal brewing data.
- Increase Accessibility and efficiency: Accommodate various field devices and communication among home brewers.

7. Conclusion

As with all software projects, there are many additional features we wanted to design and implement, but never completed due to time. Despite this, we believe the functionality in the system allows users to accomplish their home brewing goals and have fun at the same time.

Additionally, the design described in this document allows the brewers buddy website to function in a scalable and secure manner while also allowing social interaction. It accomplishes the goals we set out to achieve and would allow the system to expand in the future.