Medications Shortages Dashboard

Project Plan
Spring 2014

Spectrum Health Contact
Jeff McConnell

Team Members
Alex Lockwood
Alex Seling
Cameron Keif

Table of Contents

1. Project Overview 3
2. Functional Specifications 4
    - 2.1. Drug Collection 4
        - 2.1.1. CSV Import 4
        - 2.1.2. Manual Drug Entry 4
    - 2.2. Shortage Collection 4
        - 2.2.1. Automated RSS Feed Parsing 4
        - 2.2.2. Manual Shortage Entry 5
    - 2.3. Watch List 5
    - 2.4. Drug Detail View 5
        - 2.4.1. Notes/Collaboration 5
    - 2.5. User Privileges 5
3. Design Specifications 6
    - 3.1. User Interface 6
        - 3.1.1. Current Shortages 6
        - 3.1.2. Drug Detail View 7
        - 3.1.3. Feed Management 9
4. Technical Specifications 10
    - 4.1. System Architecture 10
    - 4.2. System Components 10
        - 4.2.1. Software Technologies 11
        - 4.2.2. Development Environment 11
        - 4.2.3. Database Schema 11
5. Testing 12
    - 5.1. Microsoft Unit Testing Framework 12
    - 5.2. Manual testing 13
6. Risks 15
7. Schedule 16

1. Project Overview

Spectrum Health is a healthcare organization which is headquartered in Grand Rapids, MI. To work effectively, the pharmacists at Spectrum Health must be able to track, visualize, and analyze medication shortages in order to make decisions to properly manage them. Currently, the pharmacists do not have any centralized system to provide this information, and must visit several third-party websites. This application, the Medication Shortages Dashboard, aims to collect the information from these external sites, and display it in a clean, easily digestible way, so users can spend less time searching for this information, and more time utilizing it.

The goal of the Medications Shortages Dashboard is to create a means to more effectively manage and analyze medication shortages, as well as provide a communication platform to discuss those shortages. The application will be used primarily by pharmacists on a daily basis and will enable them to make educated decisions regarding their inventory, based on current medication shortages. They might choose to use an alternative medication or treatment, seek out additional stock, or simply weather the shortage, based on the severity of the medication's shortage. Users will be able to create a watch-list to track particularly important drugs, and the application will provide a means of collaboration amongst pharmacists at Spectrum Health through comments on each drug's detailed view page.

2. Functional Specifications

The functional specifications section will provide a brief overview of the features and functionality provided by the Medications Shortages Dashboard. The primary function of the application is to aggregate drug and shortage information and display it in a user friendly way. In addition, the application will facilitate conversation between pharmacy staff members.

2.1 Drug Collection

The application is able to import and parse information about each drug into the database. In the future, this information will be provided by internal Spectrum Health databases, however currently this information is not readily available. As an alternative for development purposes, the drug information is provided in files from Spectrum Health.

2.1.1 CSV Import

The formulary files containing drug information are in CSV (comma separated value) files. Each drug entry in the file contains the National Drug Code (NDC), dosage, brand name, generic name, and a description of the drug. The application will utilize a parser that populates the database with these drugs.

2.1.2 Manual Drug Entry

The application will allow administrators to manually enter drug information. This manual entry will allow a user to populate the database with information that may not have been included during a csv import. The manually entered entries are treated the same way as drugs generated from csv files.

2.2 Shortage Collection

The application must be able to import and parse information about each medication shortage into the database, to be accessed by the rest of the application. The information comes from various sources.

2.2.1 Automated Feed Parsing

Every hour, the application will gather data from an RSS feed, parse it, and enter it into the database. The feed that the application uses to collect this information is from American Society of Health-System Pharmacists (ASHP).

2.2.2 Manual Shortage Entry

In some cases a user may need to manually enter drug shortage information into the system. Therefore, the application will allow high level users to enter information about a shortage. The manually created shortages will be treated the same way as shortages created by the automatic feed parsing.

2.3 Watch List

The application will provide each user with a personalized drug watch list. The user has the ability to add medications to their watch list, which will appear first on the home page.

2.4 Drug Detail View

Each drug will have a page which will show basic information about the medication, a graph of the medication's status history, and messages from users.

2.4.1 Notes/Collaboration

Pharmacy staff members and administrators will be able to post messages to each drug's detail page in order to discuss shortages. Drugs with comments that the user has not yet seen will be flagged with an icon indicating that there are unread messages.

2.5 User Privileges

Due to the business needs and processes of Spectrum Health, the application only allows certain users to make decisions about how a shortage impacts Spectrum Health. Thus, there are four levels of users in the application:

- Visitors (anyone who is not signed in) will be able to view available shortage information.
- Users will be able to have watch lists, but will not be able to see any messages on the drug detail view.
- Pharmacy staff will also have watch lists, but will be able to use messaging features.
- Pharmacy administrators will have all of the same abilities as pharmacy staff, but will also be able to create drugs and update shortage information.

3. Design Specifications

The design specifications section provides an overview of the user interface and design of the Medications Shortages Dashboard. The core of the application is the current shortages page, which shows all of the drugs, their current status, and a link to each drug's detail page. Administrators have several other pages which allow them to manage the drug and shortage information.

3.1 User Interface

3.1.1 Current Shortages

The current shortages page is the main home screen of the application. This page shows all of the drugs in the system, and their current status. The first 3 columns organize the drugs by their status, and the fourth column shows the most recently updated drugs. Each drug is color coded by the severity of its shortage status, using the following scheme:

- Green (Good): The shortage will not impact medical practice
- Yellow (Warning): The shortage may have an impact on medical practice
- Red (Severe): The shortage will definitely have an impact on medical practice

Users can log in using Windows authentication using the Login button in the top right. If a user is logged in, drugs with yellow stars indicate watch listed drugs, and will appear before items not on the watch list in their column (with the exception of the Recent column, as it is sorted by the time the drug was last updated). The star icon can be clicked to add and remove drugs from the user's watch list. Since the database contains thousands of drugs, each column only displays five drugs at a time. Controls below each column allow the user to view the next or previous subset of drugs. The search bar in the top right corner allows users to filter all of the drugs.

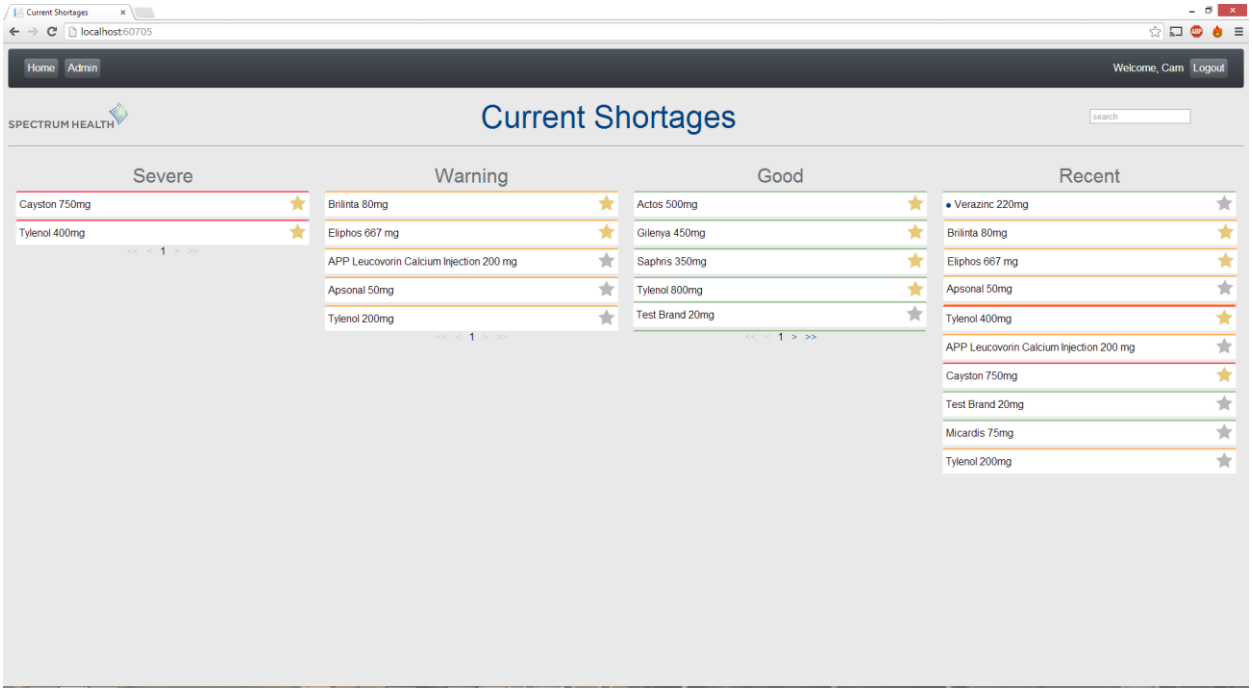

Figure 1: Current Shortages Page

3.1.2 Drug Detail View

The application will allow users to view additional details about each drug. This detail view allows users to see current shortage status, shortage history/memos, a graph of previous shortage history, and some additional information about the drug. While on a drug detail page, logged in users may add notes about a specific drug. Only pharmacy staff and administrators can see posted messages. Finally, pharmacy administrators can create, edit, or delete shortages for that drug from this page.

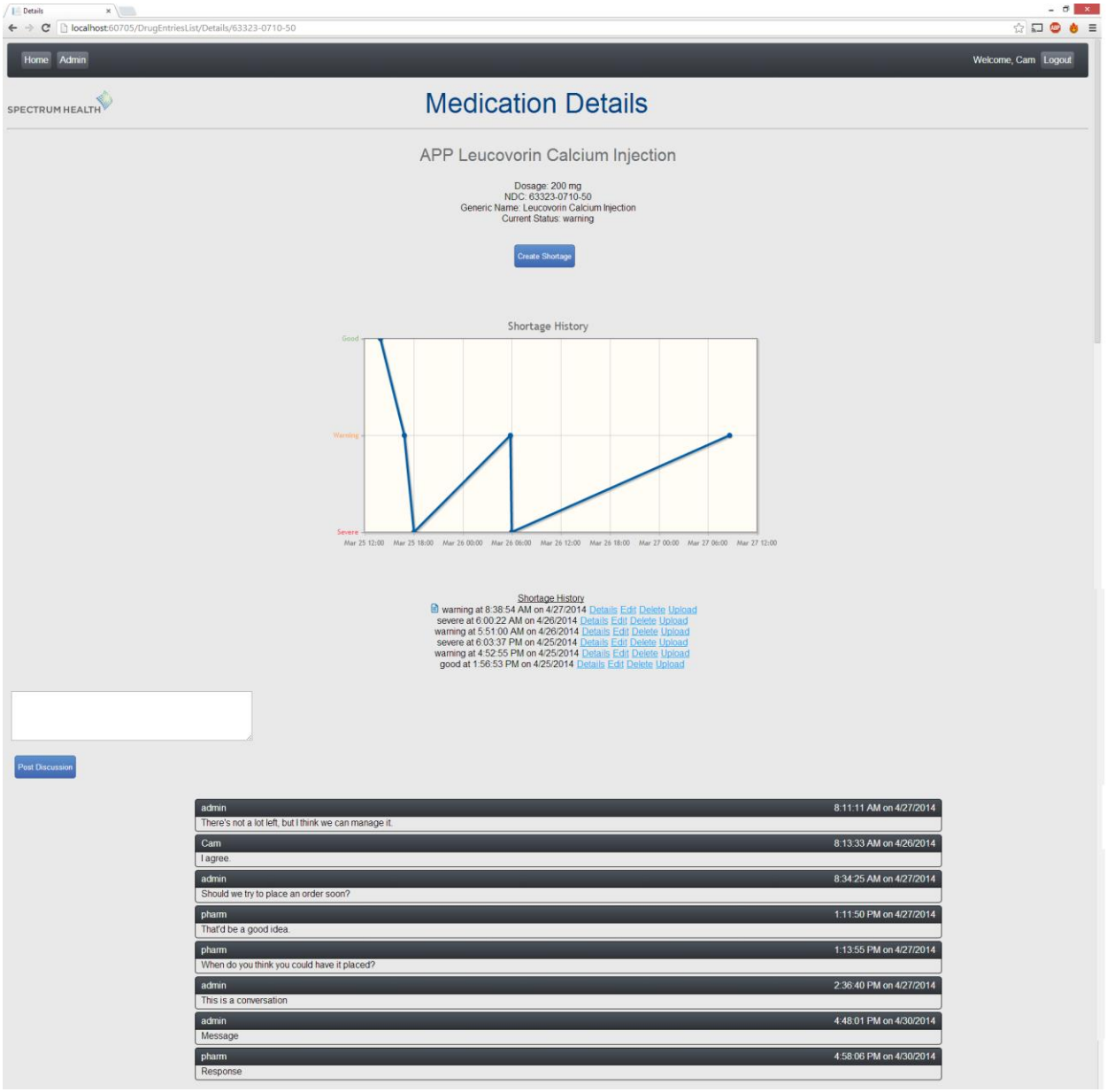

Figure 2: Drug Detail View

3.1.3 Feed Management

If a user is logged in as a pharmacy administrator, they will have access to the Feed Management page. This page pulls information from the ASHP current shortages feed and summarizes it, allowing administrators to keep up to date on the most recent shortages. If a product carried by Spectrum Health is affected, the user has the ability to directly update shortage status from this page by clicking the Create Shortage button. If the administrator is no longer interested in an item from the feed, they can click the Ignore button to prevent that notification from appearing again. All ignored items can be viewed and removed from the ignored list by clicking Ignored Notifications.

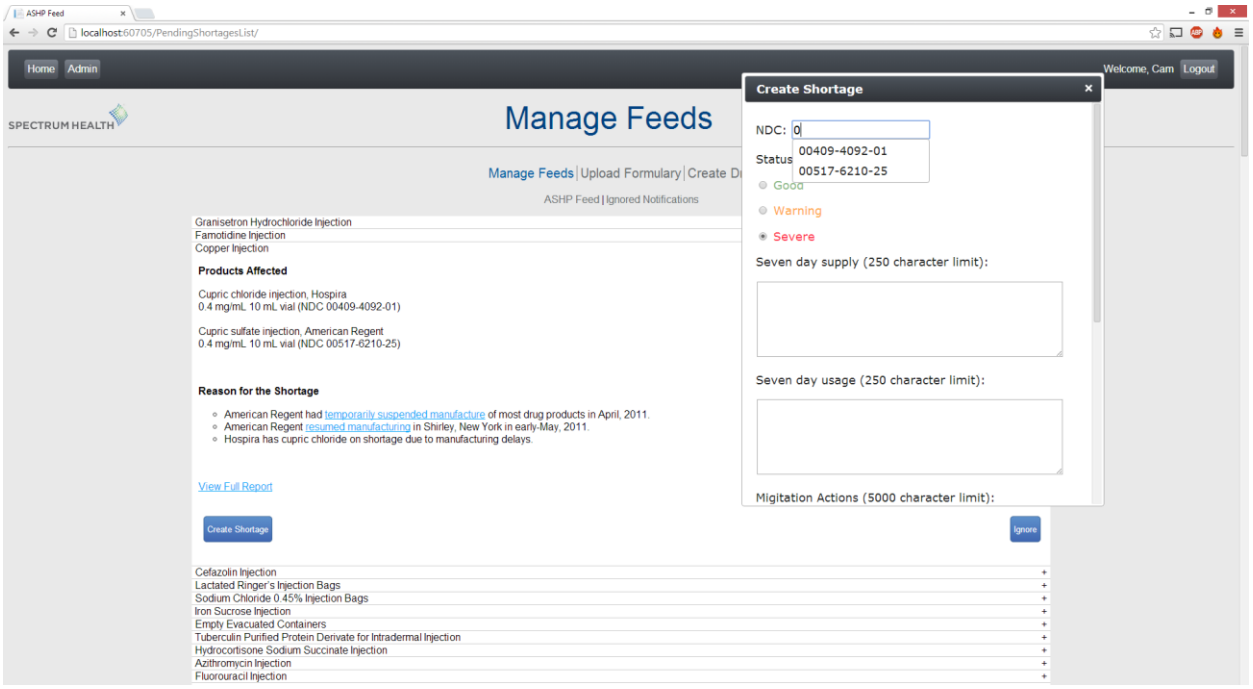

Figure 3: Feed Management Page

4. Technical Specifications

The technical specifications section focuses on the technical details of the system. It contains an explanation of the system architecture and database schema. Additionally, the technologies and development environment used to build this application are described.

4.1 System Architecture

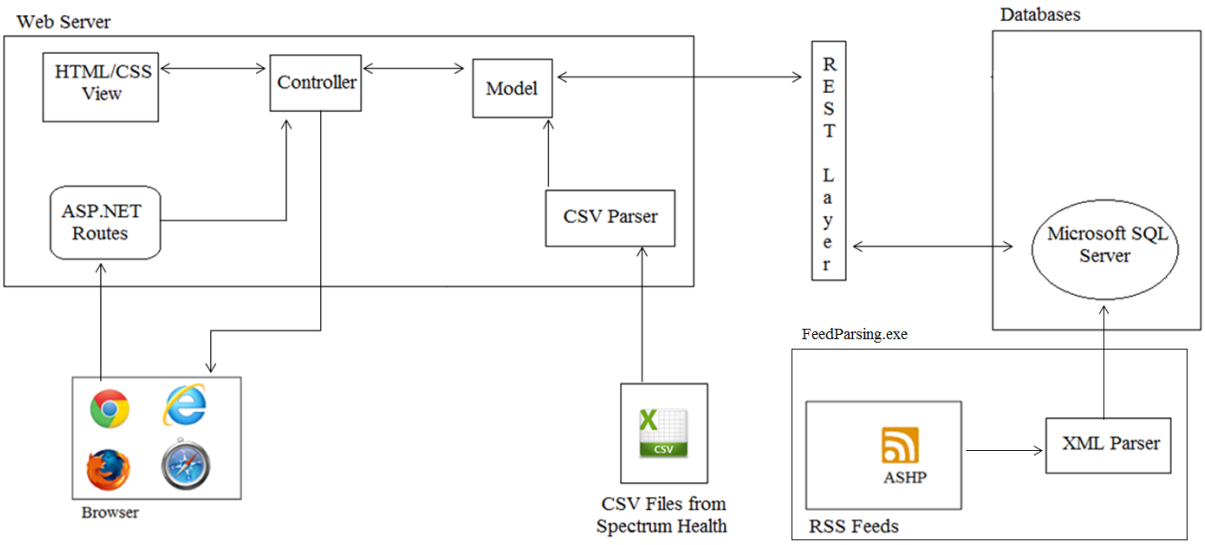

Figure 4: System Architecture

Figure 4 provides a high level overview of the application's structure. The application follows the MVC software design pattern. The user navigates to the webpage, and ASP.NET routes their request to the appropriate controller. The C# code in the controller then creates the appropriate view (HTML/CSS/JavaScript), by using data provided by a model, which interacts with the Microsoft SQL database through a REST layer.

The ASHP feed is parsed by the XML Parser every hour, which then sends new entries into the database through a model. The model will then populate the database through the REST layer. The CSV files will be handled the same way, the only difference being that they are not automatically parsed, but rather an administrator will import the file whenever they want.

4.2 System Components

The application was built primarily with Microsoft technologies, because they are currently supported by the teams at Spectrum Health.

4.2.1 Software Technologies

- ASP.Net Framework 4.5.0
- Language: C#
- Razor View Engine
- JavaScript & jQuery
- HTML/CSS
- SQL Server 2008

4.2.2 Development Environment

The application is written using Visual Studio 2012 Ultimate on Windows 7 and Windows 8 machines, and the server is run on Windows Server 2008. Source control is provided by Git in a private repository on GitHub. Unit testing is done using Microsoft Unit Testing Framework. StyleCop is used to maintain consistent coding standards.

4.2.3 Database Structure

The database is set up as follows (bold/underlined fields are primary keys):

Table: DRUG_VISIT

ID	USERNAME	NDC	DATE

Table: DRUGS

NDC	DOSAGE	BRAND_ NAME	GENERIC_ NAME	CURRENT_ STATUS	LAST_ UPDATED	LAST_POST	DESCRIPTION

Table: IGNORED_PENDING_SHORTAGES

ID	**USERNAME**	DRUG_NAME

Table: MESSAGE

ID	NDC	USER	DATE	TEXT

Table: PENDING_SHORTAGES

ID	DRUG_NAME	UPLOAD_TIME

Table: SHORTAGE

ID	NDC	DATE TIME	SEVEN_ DAY_ SUPPLY	SEVEN_ DAY_ USAGE	MITIGATION_ ACTIONS	GREY_ MARKET_ VENDOR
BACKORDER_ INFORMATION			EXPECTED_ RESOLUTION_ DATE		MEMO	

Table: USER

USERNAME	PASSWORD	TYPE

Table: WATCHLIST_ITEM

USERNAME	**NDC**

5. Testing

The testing section provides an overview of the testing methodologies used for the Medications Shortages Dashboard. The team utilizes unit testing to ensure system stability, and manual exploratory testing in order to discover bugs.

5.1 Microsoft Unit Testing Framework

Microsoft Visual Studio 2012 contains built-in testing tools, all of which belong to the *Microsoft.VisualStudio.TestTools.UnitTesting* namespace. This is set up using an additional test project in tandem with the project to be tested. Tests are created in any number of test classes and methods, all directly interacting with specific classes in their methods.

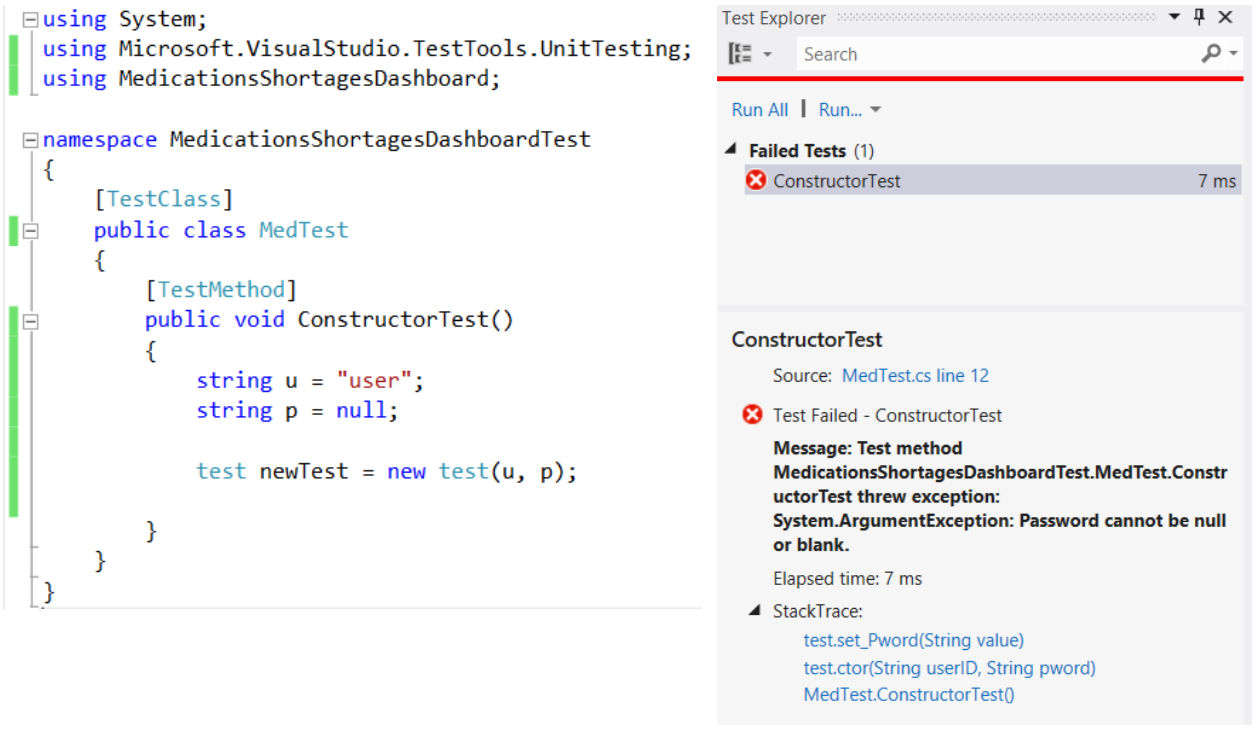

Figure 5: Testing class and TestingExplorer

As shown in Figure 5, test classes can be extremely simple, and are always decorated with *TestClass* and *TestMethod* attributes. In this example, it creates a *test* class with two parameters, and within said class, it tests those input parameters. In this case, the user string is allowed, but the password is not allowed to be null, so it displays said error in the *Test Explorer*.

Since each set of tests can be customized for each part of the project, we will be using this interface extensively, due to its simplicity, recyclability, and extensibility. However, the only limitation is testing exterior input information, as anything outside of Visual Studio cannot be directly tested. However, this will be handled by extensive exception catching in every class and manual testing of read information.

5.2 Manual Testing

Since nearly all of the data used in our project will be pulled from exterior sources, manually testing for ranges of input data will be necessary. This includes everything from testing created users and handling login security, streamed and parsed pharmaceutical feeds and any information contained within, to further exterior sources and feeds that may be used.

When the team adds new functionality, it will be tested by performing both expected and,

in some cases, unexpected workflows to try and break that feature. Old functionality will also be briefly tested to ensure new changes do not interfere with and break old code. Between unit testing and exploratory manual testing, most bugs should be found during development.

6. Risks

The risks that the team had to handle during development of the application, as well as the way the risk was mitigated are as follows:

- Shortages Feed – Where is the ASHP feed found? How can it be read?
    - The client provided the location of the feed, and an existing C# XML parsing library was used to read it
- Unsure how to set up/use Microsoft SQL Server and Microsoft Server 2008
    - The team utilized online tutorials and self-teaching to learn these technologies
- Getting the website live – How can we get the website hosted on our server?
    - The team experimented through trial and error, and using tutorials and stackoverflow managed to figure out how to publish the website.

7. Schedule

Below is a schedule of the teams progress throughout development of this application.

Week 1
Environment setup
Domain research
Getting familiar with system components

Week 2
Create Database Schema
UI Screen Mockups
Project plan initial draft

Week 3
Database/server setup
XML Parsing
CSV Parsing
Project Plan Presentation

Week 4
XML Parsing
CSV Parsing
Database/server setup

Week 5
Authentication
Automatic XML Parser
Drug Detail View

Week 6
Authentication
User Privileges
Watch List

Week 7
Alpha Presentation
Parsing for csv files

Week 8
Manual Entry of Drugs/Shortages
Messages on Drug Detail View
Admin Panel

Week 9
Drug Detail View graph
Messages on Drug Detail View
Search

Week 10
Paging
Expanded shortage creation
UI Tweaks

Week 11
Beta Presentation
Debugging (heavy manual exploratory testing)
Documentation

Week 12
Debugging
Documentation
Final UI Tweaks

Week 13
Project Video
Debugging

Week 14
Project Video
Debugging
Code cleanup

Week 15
Finalize project
Submission
Design Day