

System Design Specification For



Submitted by
Team Savvy

Team Members: William Ashman
Eric Buck
Jordan Jobs
Colin McIntosh
Christian Santarelli
Ryan Timken

Instructor: Dr. Jennifer Booker

Cycle: 3

Date Submitted: Tuesday, April 26th, 2016

Grading Rubric - Design Specification

This rubric outlines the grading criteria for this document. Note that the criteria represent a plan for grading. Change is possible, especially given the dynamic nature of this course. Any change will be applied consistently for the entire class.

Achievement	Minimal	Exemplary	Pts	Score
Content	Section(s) missing, not useful, inconsistent, or wrong.	Provides all relevant information correctly and with appropriate detail		
Introduction			10	
Architectural Description			10	
Interface Description			15	
Detailed Design			50	
Grammar and Spelling	Many serious mistakes in grammar or spelling	Grammar, punctuation, and spelling all correct	5	
Expression	Hard to follow or poor word choices	Clear and concise. A pleasure to read	5	
Tone	Tone not appropriate for technical writing	Tone is consistently professional		
Organization	Information difficult to locate	All information is easy to find and important points stand out	5	
Layout	Layout is inconsistent, visually distracting, or hinders use	Layout is attractive, consistent, and helps guide the reader		
Late Submission				
Total			100	

Table of Contents FIX THIS

- 1. Introduction**
 - 1.1. Scope
 - 1.2. Definitions, Acronyms, and Abbreviations
 - 1.3. References
 - 1.4. Requirements Traceability Matrix
- 2. Architectural Description**
 - 2.1. Architectural Overview
 - 2.2. Component Decomposition
- 3. Interface Description**
 - 3.1. User Interface
 - 3.2. Data Interface
 - 3.3. Programming Interface
- 4. Detailed Design**
 - 4.1. User Interface**
 - 4.1.1 Home Page (Logged In)
 - 4.1.2 Home Page (Not Logged In)
 - 4.1.3 Price Submission Page
 - 4.1.4 Search Results Page
 - 4.1.5 Product Information Page
 - 4.1.6 Login Page
 - 4.1.7 Signup Page
 - 4.1.8 Account Recovery Page
 - 4.1.9 Admin Page
 - 4.1.10 View User Information Page
 - 4.1.11 Receipt Edit Page
 - 4.1.12 User Settings Page
 - 4.2 Data Schema**
 - 4.2.1 Data Flow Diagram
 - 4.2.2 Entity Relationship Diagram
 - 4.3 Data Classes**
 - 4.3.1 User
 - 4.3.2 Business
 - 4.3.3 Product
 - 4.3.4 Price

Table of Contributions

The table below identifies contributors to various sections of this document.

	Section	Writing	Editing
1	Introduction	EB, JJ	RT, CS
2	Architectural Description	JJ, RT	CM, EB
3	Interface Description	CS, RT, WA	EB, JJ
4	Detailed Design	WA, CS, CM, RT, EB	JJ

1. Introduction

1.1. Scope

This document outlines the design specifications for Savvy, a web-based tool for finding localized price information. The system requirements specification document will be referenced as a source for the various design implementations.

1.2. Definitions, Acronyms, and Abbreviations

API - Application program interface is a software component that allows for programs to interact and share data with each other.

DFD- Data flow diagram is a graphical explanation of how information flows through an information system.

FAQ - Frequently Asked Questions.

GUI- Graphical user interface is the interface that the user will use to interact with the application.

Product - A product represents a good or service.

Receipt - This represents a receipt from the product(s) point of purchase.

RESTful - Representational State Transfer is a structure for web-applications that decouples front-end functionality from back-end architecture. It allows for multiple front-ends on distinct platforms to use a common back-end.

RFC - Request for comments is a publication for technical development and standards.

Tags - User submitted words or phrases used to categorize products or services.

1.3. References

A hackable text editor for the 21st Century. (n.d.). Retrieved February 15, 2016, from <https://atom.io/>

Ashman, W., Buck, E., Jobs, J., McIntosh, C., Santarelli, C., Timken, R. (2015, January 26). System Requirements Specifications for Savvy.

Bray, T. (2014, March 1). RFC 7159 - The JavaScript Object Notation (JSON) Data Interchange Format. Retrieved November 10, 2015, from <https://tools.ietf.org/html/rfc7159>.

Cascading Style Sheets home page. (n.d.). Retrieved February 15, 2016, from <https://www.w3.org/Style/CSS/>

Google Maps APIs | Google Developers. (n.d.). Retrieved November 7, 2015, from <https://developers.google.com/maps/?hl=en>.

Google Places API | Google Developers. (n.d.). Retrieved November 10, 2015, from <https://developers.google.com/places/>.

HTML5. (n.d.). Retrieved February 15, 2016, from <https://www.w3.org/TR/html5/>

PyCharm. (n.d.). Retrieved February 15, 2016, from <https://www.jetbrains.com/pycharm/>

MongoDB 3.2. (n.d.). Retrieved February 15, 2016, from <https://www.mongodb.org/>

JavaScript. (n.d.). Retrieved February 15, 2016, from <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

What's New In Python 3.5. (n.d.). Retrieved February 15, 2016, from <https://docs.python.org/3/whatsnew/3.5.html>

1.4. Requirements Traceability Matrix

This section maps the relationship between requirement statements and detailed design entities. As such it shows how requirements are covered by the design, and demonstrates the purpose for which design entity exists.

The values in the cells of tables 1.1 and 1.2 show which requirements provide the purpose for each entity. The cell values are:

- **Blank** – the design entity does not implement any of that requirement
- **P** for Primary - the design entity implements all or most of the requirement
- **S** for Secondary – the design entity implements a smaller but essential part of the requirement

Detailed Design Entity Identifier Table 1.1

Req.	4.1.1	4.1.2	4.1.3	4.1.4	4.1.5	4.1.6	4.1.7	4.1.8	4.1.9	4.1.10	4.1.11	4.1.12	4.1.13	4.1.14
3.2.1.1						S	P							
3.2.1.2							P							
3.2.1.3										P				
3.2.1.4						P								
3.2.1.5								S	P					
3.2.1.6									P			S		
3.2.1.7								P						
3.2.2.1										S	P	S		
3.2.2.2											P			
3.2.2.3											P			
3.2.2.4											P			
3.2.3.1			P											
3.2.3.2			P											
3.2.3.3													P	
3.2.3.4													P	
3.2.3.5			P											
3.2.3.6				P	S									
3.2.3.7				P	S									
3.2.3.8			P		S									
3.2.3.9					P									
3.2.4.1	P			S										
3.2.4.2	P			S										
3.2.4.3	P			S										
3.2.4.4	P			S										
3.2.5.1					P									
3.2.5.2		S		P										
3.2.5.3					P									
3.2.5.4					P									
3.2.5.5					P									
3.2.6.1														P
3.2.6.2														P
3.2.7.1														
3.2.7.2														

Table 1.1 – Traceability between Requirements and User Interface Design Entities

Detailed Design Entity Identifier Table 1.2

Req.	4.2.1	4.2.2	4.3.1	4.3.2	4.3.3	4.3.4
3.2.1.1	P	P	P			
3.2.1.2	P	P	P			P
3.2.1.3	P	P	P			
3.2.1.4	P	P	P			
3.2.2.1	P	P	P			
3.2.2.2	P	P	P			P
3.2.2.3	P	P	P			
3.2.2.4	P	P	P			
3.2.3.1	P	P		P		P
3.2.3.2	P	P		P		P
3.2.3.3	P	P		P		
3.2.3.4	P	P		P		
3.2.3.5	P	P		P	P	
3.2.3.6	P	P		P	P	
3.2.3.7	P	P		P	P	
3.2.3.8	P	P		P		P
3.2.3.9	P	P		P		
3.2.4.1	P	P		P	P	
3.2.4.2	P	P		P		
3.2.4.3	P	P		P		
3.2.4.4	P	P				
3.2.5.1	P	P				
3.2.5.2	P	P				P
3.2.5.3	P	P				
3.2.5.4	P	P				
3.2.5.5	P	P				
3.2.6.1	P	P				
3.2.6.2	P	P				
3.2.7.1	P	P				
3.2.7.2	P	P				

Table 1.2 – Traceability between Requirements and User Interface Design Entities

2. Architectural Description

As shown in Figure 2, the system's design layers are the front-end interface, RESTful API, and the data storage. The interface layer will include the front-end interface, in which the user can create a user account and submit or search for price data. The user interface is further detailed in sections 4.1.1 through 4.1.14. The storage design is detailed in section 3.3.

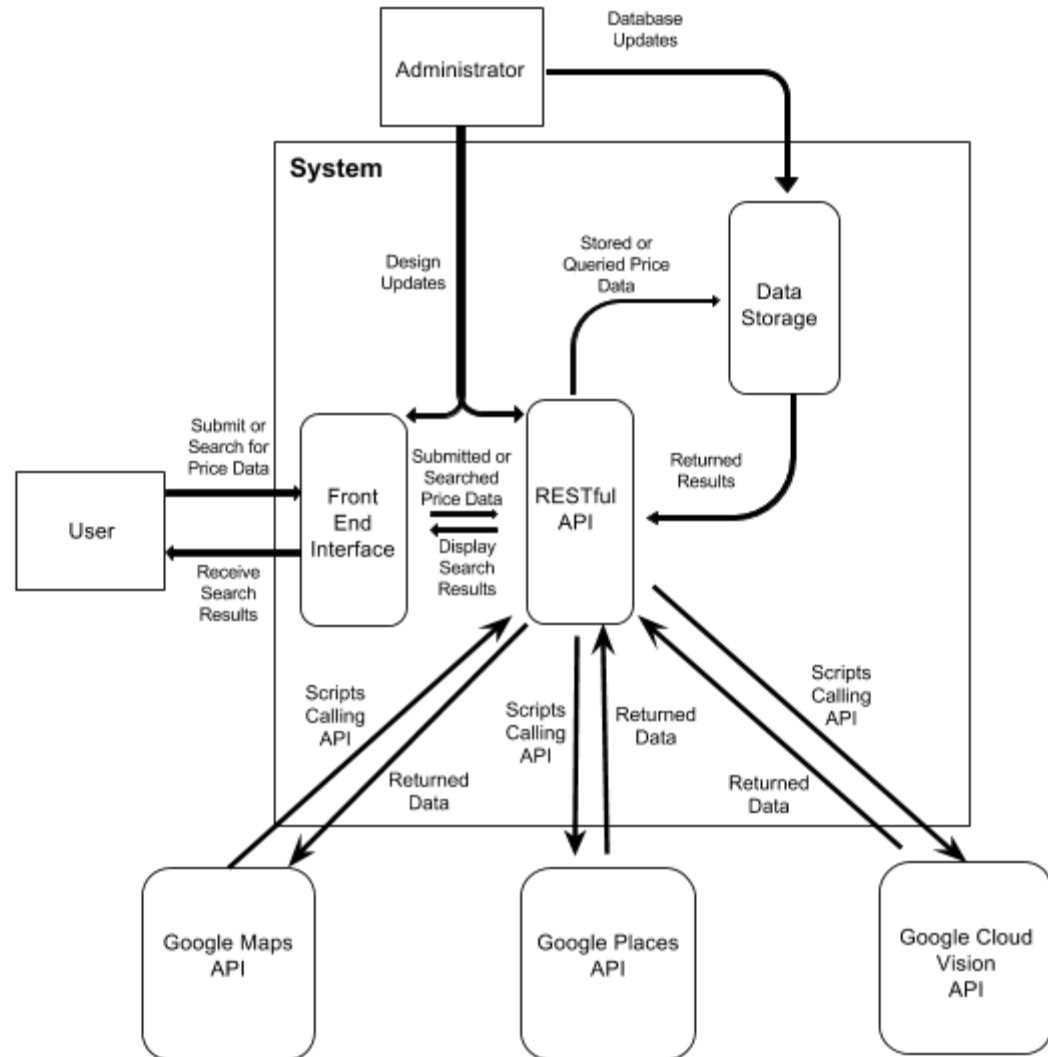


Figure 2: System Architecture Overview

2.1. Front-End Interface

The front-end interface is available through a web page which is accessible via desktop or

mobile browsers. The front-end interface will make calls to the RESTful API based on actions performed by users or administrators. The front-end interfaces are further detailed in sections 4.1.1 to 4.1.14.

2.2. RESTful API

Savvy will feature an authenticated and unauthenticated RESTful API for use by Savvy developers and developers outside of Savvy. The RESTful API is used for all communication between the front-end and back-end. The RESTful API is depicted in Figure 2.

2.3. Data Storage

Savvy utilizes cloud data storage for all of its databases. Cloud storage ensures that all of Savvy's databases are redundant, highly available, and easily backed up. The RESTful API is depicted in Figure 2.

2.4. Google Places

Google Places is a public API (Google Places API, n.d.) that allows developers to retrieve business names, business addresses and business contact information from Google's databases. The Google Places API is further detailed in section 3.2.

2.5. Google Maps

Google Maps allows web developers to embed maps in their own webpages (Google Maps API, n.d.). Savvy utilizes this feature in Google Maps to show users where businesses are located. The Google Maps API is further detailed in section 3.2.

2.6. Google Cloud Vision API

Google's Cloud Vision API allows web developers to convert text from images into plaintext via optical character recognition. Savvy will forward user uploaded receipts to the Cloud Vision API and the API will return the text that was printed on the receipt. The text will be automatically entered onto the price submission page.

3. Interface Description

3.1. User Interface

When users browse to the Savvy website, they are greeted by the Savvy homepage as shown in Figure 3, and it is further detailed in Figure 4.1.1. The homepage gives users the option to search for products and services or navigate to the following pages: Login, Signup, Submit. Users must be logged-in in order to submit and view price information. If a user who is not logged in performs a search, the results page will display products or services but no price information.

The results page will display products related to the user's query and allow users to navigate to individual product and service pages that display more price metrics as well as relevant business information. The results page is shown in Figure 4.1.4.

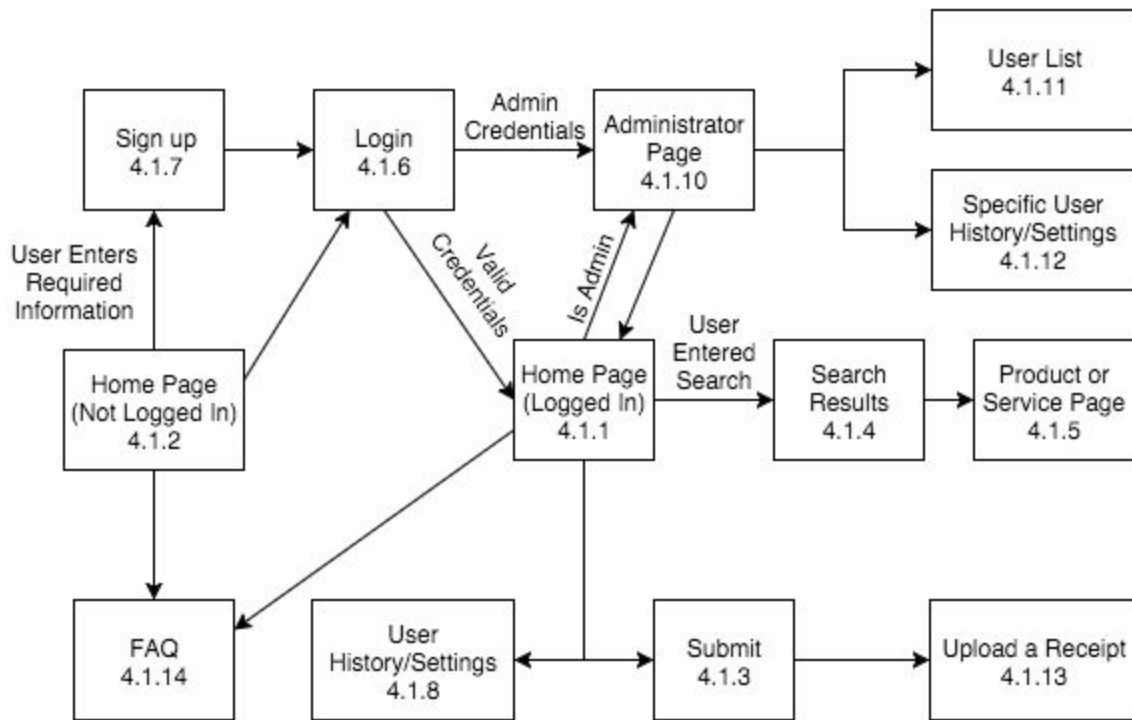


Figure 3: Screen Flow Diagram

3.2. Data Interface

Google Maps - The system will pull a business's location data from Google Maps as it pertains to a product or service. This data will be queried as a component of the RESTful API layer in Figure 2 (Google Maps API, n.d.). The information will be provided via Python scripts between the RESTful API layer and the Google Maps API.

Google Places - The system will pull business details from Google Places as the business pertains to a product or service. This data will be queried as a component of the RESTful API layer in Figure 2 (Google Places API, n.d.). The information will be provided via Python scripts between the RESTful API layer and the Google Places API.

Google Cloud Vision - The system will send user uploaded receipts to Google Cloud Vision API and receive data in the form of text. This data will be sent and received as a component of the RESTful API layer in Figure 2 (Google Cloud Vision API, n.d.). The information will be provided via Python scripts between the RESTful API layer and the Google Cloud Vision API.

3.3. Programming Interface

The system will provide an Application Programming Interface (API) that will be used by the system front-end and developers. The API will be served over an HTTP REST

connection. The API will serve and accept data in JSON format as defined by RFC 7159 (Bray, 2014).

Savvy's front-end utilizes JavaScript, HTML5, CSS3, and all of the code is written using GitHub Atom IDE version 1.1.0. Python version 3.5 and PyCharm version 5 are used for back-end development. Savvy's back-end employs MongoDB as its database. References for all of the aforementioned software and frameworks can be found in section 1.3.

4. Detailed Design

4.1. User Interface

4.1.1 Home Page (Logged In) (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirement 3.2.4.1 through 3.2.4.4.

Description: Figure 4 will be the view the user sees if they have already logged in to Savvy. The view will display easy access to the “search” functionality as well as the user submission of a new price. This page will also include a description of how to use features within Savvy. Figure 4 shows the navigation menu at the top of the page and the footer at the bottom of the page. These are shown in the same location on every page of the user interface. If the user is logged in and has admin privileges, a special link will appear on the navigation menu that will allow them to navigate to the administrator’s only page.

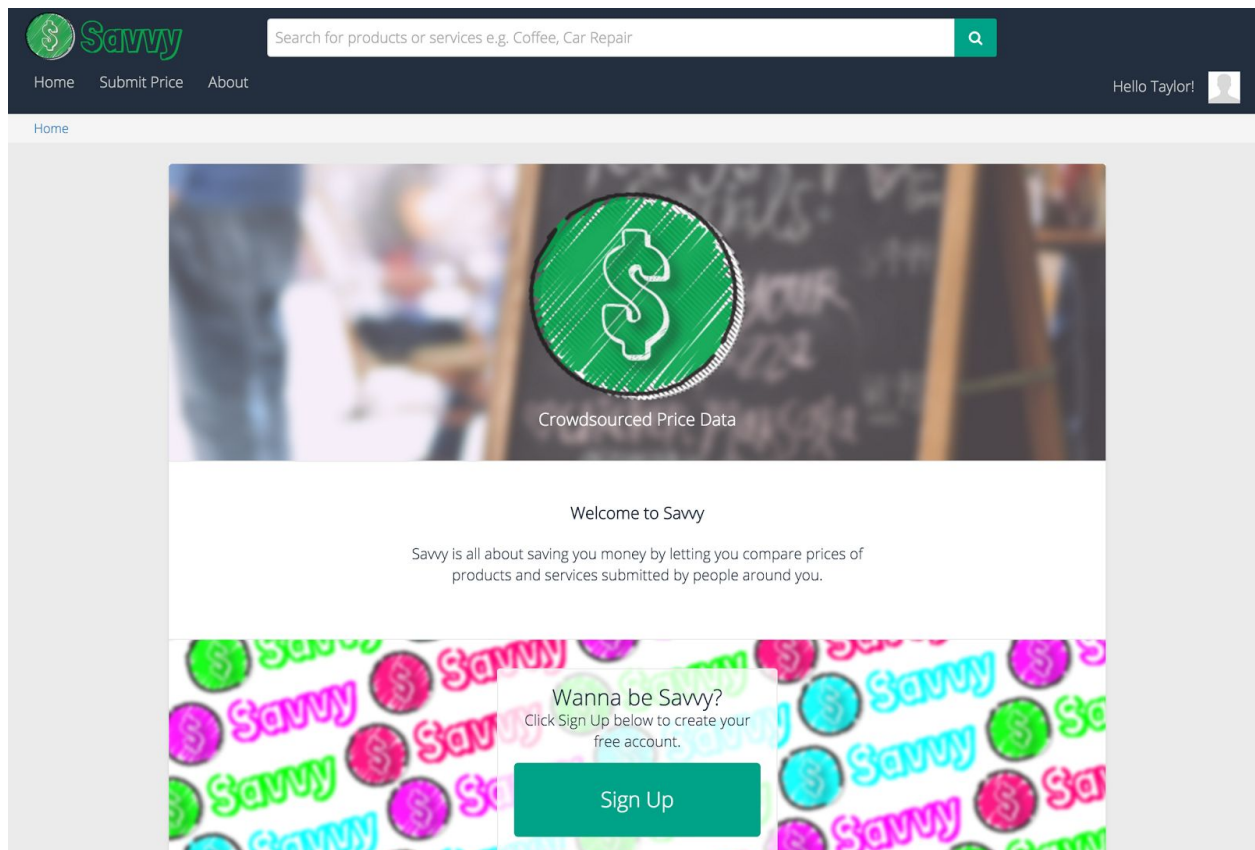


Figure 4: Home Page Logged in

4.1.2 Home Page (Not Logged In) (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirement 3.2.5.1.

Description: Figure 5 will be the view the user sees if they have not already logged in to Savvy. The view will display easy access to the “search” functionality as well as a button to transfer the user to the signup page. As you can see the word ‘milk,’ is being searched for and the results will be shown in figure 7. If the user has already registered with Savvy they will also have easy access to a login link. Figure 5 will also include a description of how to use features within Savvy. Figure 5 will include the navigation and footer as is standard for every page.

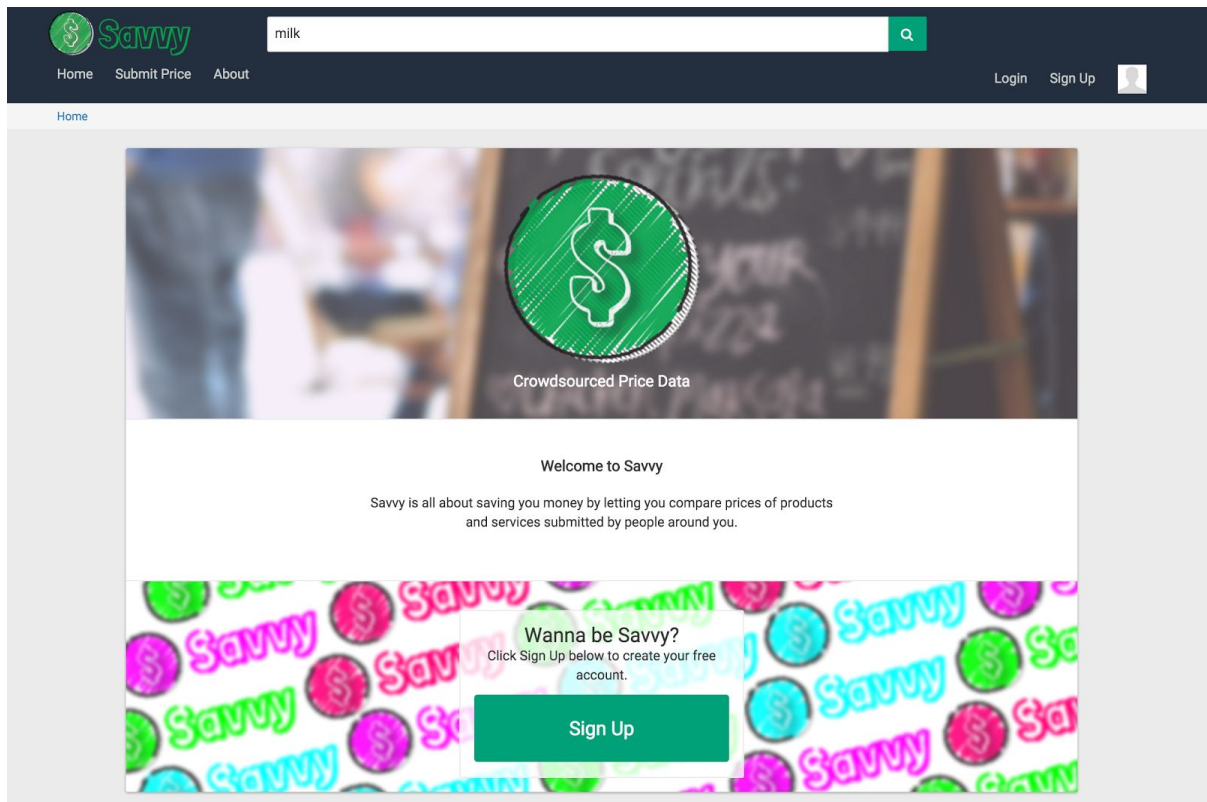


Figure 5: Home Page (Not Logged in)

4.1.3 Price Submission Page (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirement 3.2.3.1, 3.2.3.2, 3.2.3.5, and 3.2.3.8.

Description: Figure 6 is the page where the user can submit a price of a product or service. The user must complete this form before submitting a price to the site. Here we ask the user for information such as the brand, product name, place of purchase, price, and an optional photo of the product and tags that are associated with the product. If the user needs to exit this page they can use the breadcrumbs below the navigation bar.

Home Submit Price About

Search for products or services e.g. Coffee, Car Repair

Hello Taylor!

Home / Submit

Submit a Price

Submit a Single Price Upload a Receipt

Complete the required (*) fields below then click Submit to add your product or service to Savvy.

Product Name *

The name of the product, e.g. 12oz. Coke Can

Place of Purchase *

Where you bought the product [Enter Custom Address](#)

Price Paid *

\$ The price you paid

You can add tags here. Click add after entering each tag. You may submit more than one tag.

Add A Tag

Enter a tag [Add](#)

Upload a picture of the product

[Choose File](#) No file chosen

[Submit](#)

Figure 6: Submit a Price

4.1.4 Search Results Page (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirement 3.2.3.6, 3.2.3.7, 3.2.4.1 through 3.2.4.4, and 3.2.5.2.

Description: Figure 7 will display the results returned from the database when a user performs a search. The database entities are detailed in section 4.2 and 3.2. Figure 7 will display a photo of the product (if a user submits an image), product name, lowest price, and average price. The product name and photo is a link to the product information page where the user can view more pricing and information about the product. The user can also filter the list of results by business name, price, and product name. There is navigation at the bottom if there is more than one page of results for the user's query. If a user has not registered to the site, they will only see products, but no prices.

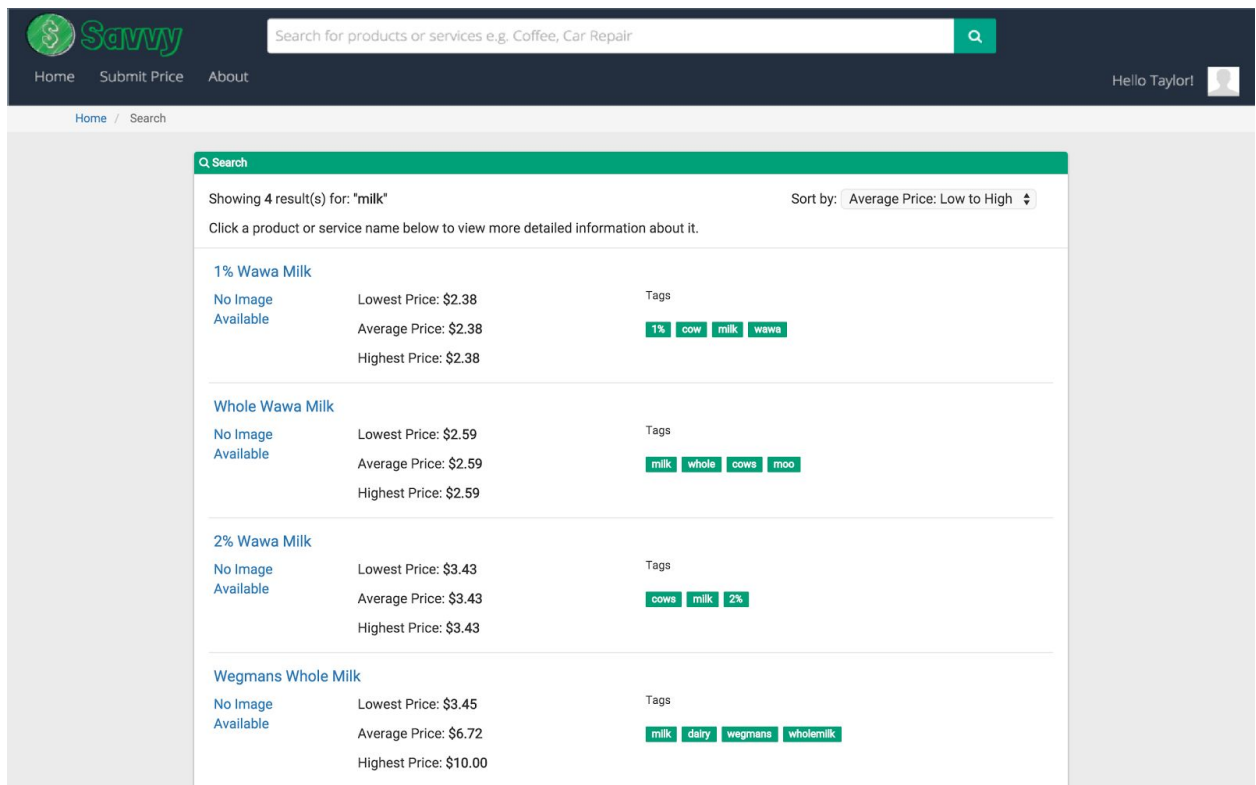


Figure 7: Search Results Page

4.1.5 Product Information Page (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirement 3.2.3.6 through 3.2.3.9, 3.2.5.1, and 3.2.5.3 through 3.2.5.5.

Description: Figure 8 will display detailed information about a product after it has been selected in Figure 7. The product's name will be displayed at the top along with its current average price and other price metrics. An image of the product will also be displayed next to the name and the price metrics. In the middle of the page, other price metrics (such as prices over a period of time) will be displayed graphically. On the bottom of the page, more data will be displayed in tabular format. This data will include “related brands” that will show competitive price information as well as a “pricing history” listing of recently submitted prices for the product. There is also an up and down arrow on the page to let the user report if the pricing information on the page was helpful to them.

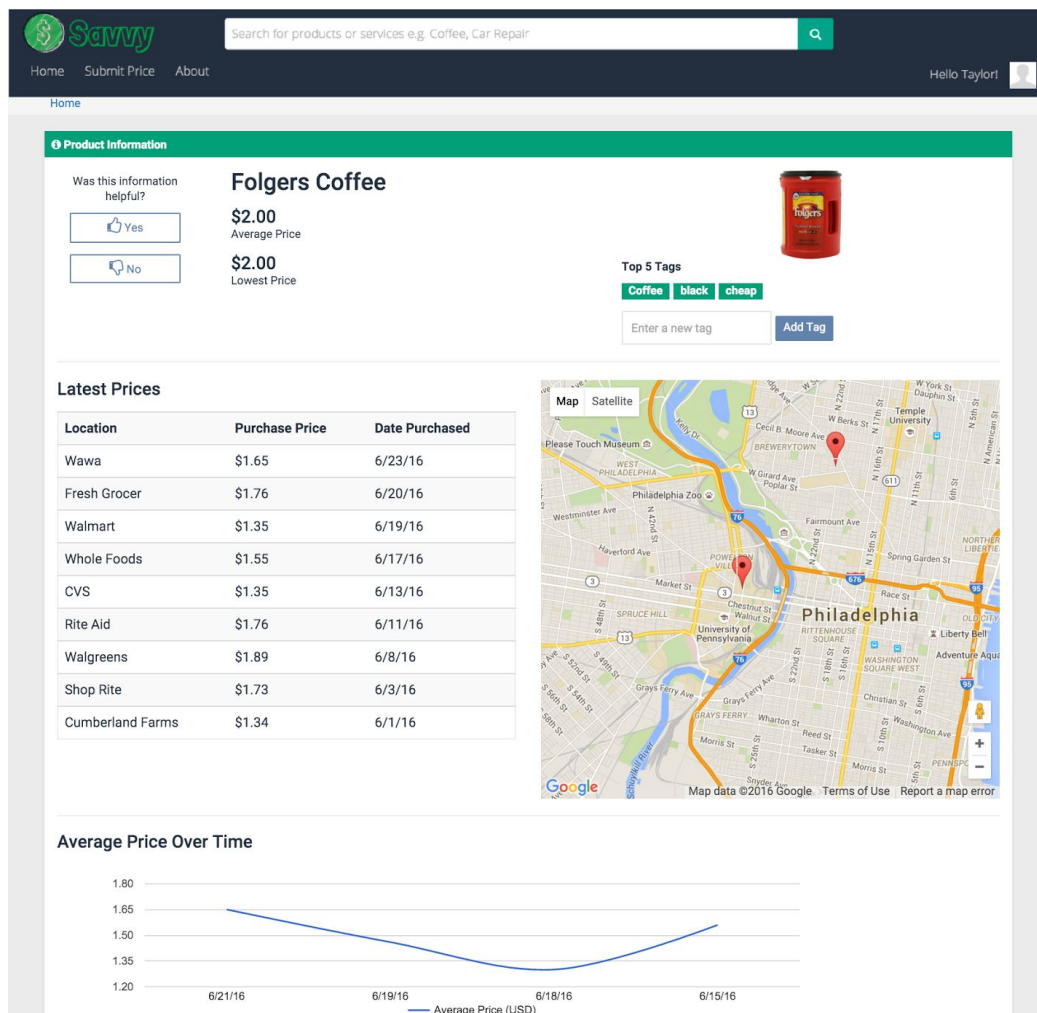


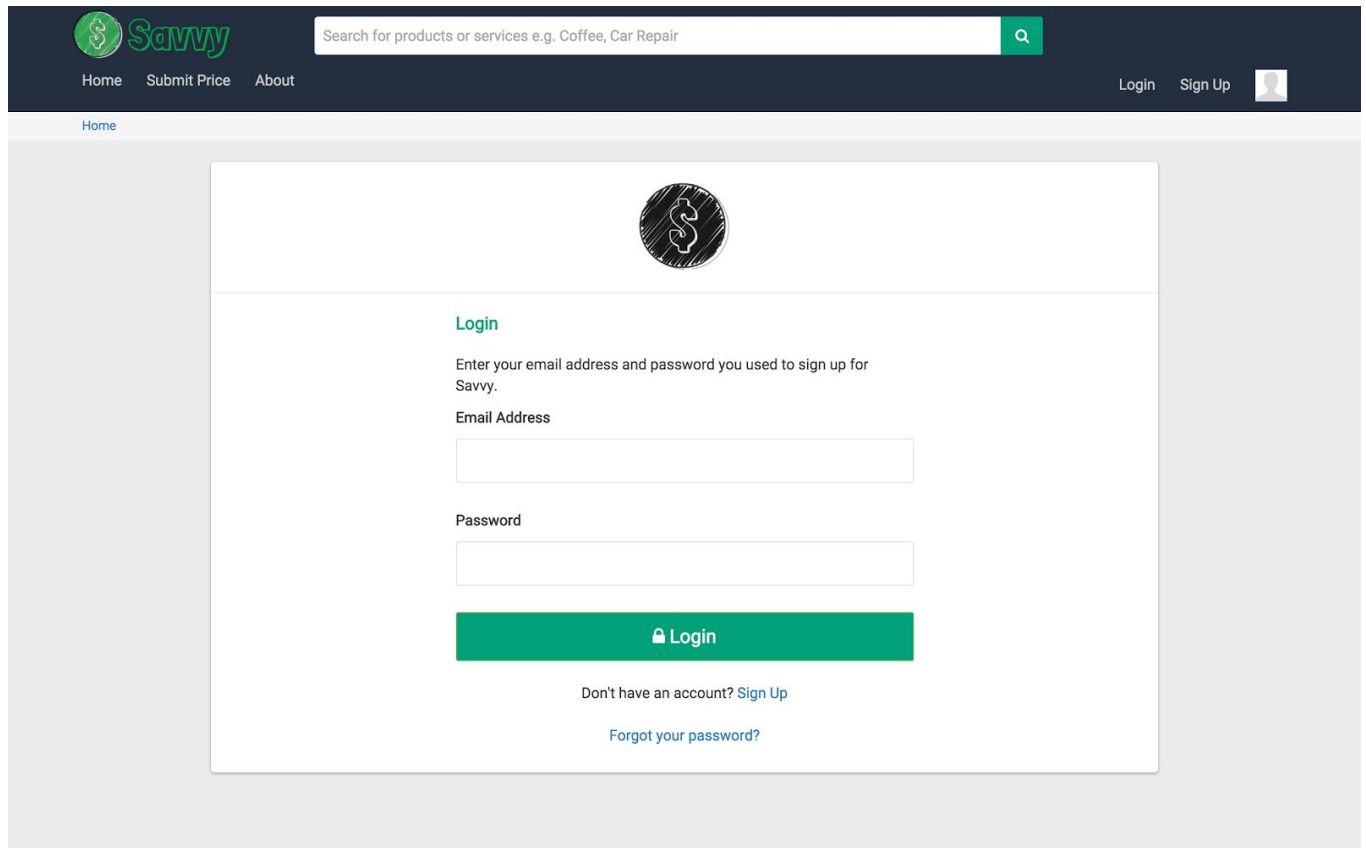
Figure 8: Product Information Page

4.1.6 Login Page (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirement 3.2.1.1 and 3.2.1.4.

Description: Figure 9 is the page the user will see when they click “Login” on the sites homepage. The user will input their email address (username) and password they used to register for the site. If the user lands on this page accidentally, there will be a link to the “Signup” page below the form, the signup page is show in figure 10. There is a link to the “Account Recovery” page in the case the user forgot their password, this page is shown in figure 12.



The screenshot displays the login interface for 'Savvy'. At the top, a dark blue header contains the 'Savvy' logo (a green circle with a white dollar sign) on the left, a search bar with the placeholder text 'Search for products or services e.g. Coffee, Car Repair' in the center, and navigation links 'Home', 'Submit Price', and 'About' on the right. Below the header, a light gray sidebar on the left has a 'Home' link. The main content area is white and features a circular icon with a dollar sign at the top. Below the icon, the word 'Login' is written in green. A prompt reads: 'Enter your email address and password you used to sign up for Savvy.' There are two input fields: 'Email Address' and 'Password'. A green 'Login' button with a lock icon is positioned below the fields. At the bottom, there are two links: 'Don't have an account? Sign Up' and 'Forgot your password?'.

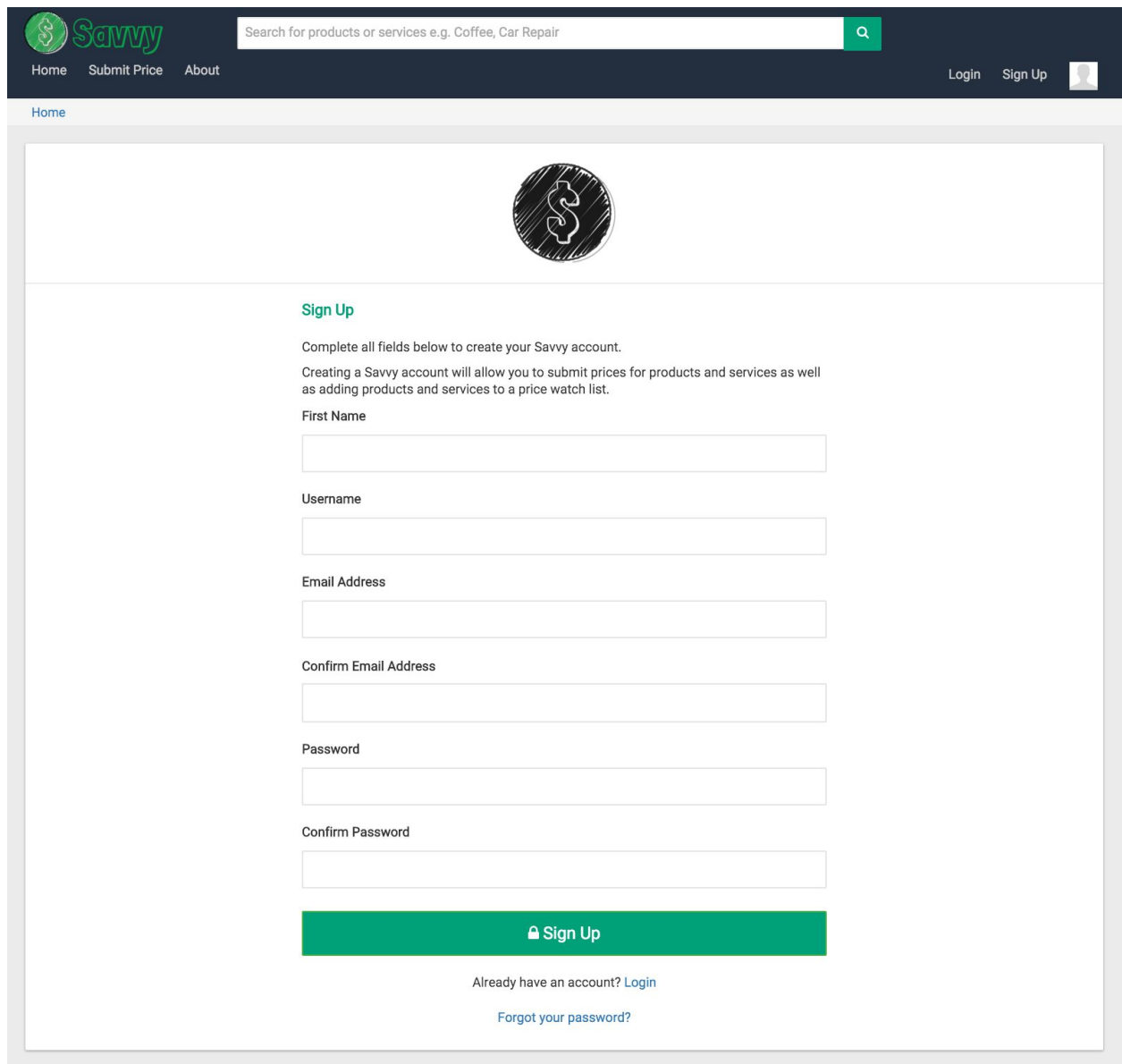
Figure 9: Login Page

4.1.7 Signup Page (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirements 3.2.1.1 and 3.2.1.2.

Description: The user will see Figure 10 when they click on “Signup” on the homepage in Figure 4.1.1. Here the user will input their first name, username, email address, and a password of their choice. If the user already created an account and is on this page by mistake there is a link to the “Login” page on the top right in the navigation bar.



The screenshot shows the Savvy website's Signup page. The header is dark blue with the Savvy logo (a green circle with a white dollar sign) on the left. To the right of the logo is a search bar with the placeholder text "Search for products or services e.g. Coffee, Car Repair" and a green search button. Below the header is a navigation bar with links: Home, Submit Price, About, Login, Sign Up, and a user profile icon. The main content area is white and features a large, stylized dollar sign icon at the top. Below the icon, the text "Sign Up" is displayed in green. The instructions state: "Complete all fields below to create your Savvy account. Creating a Savvy account will allow you to submit prices for products and services as well as adding products and services to a price watch list." The form fields are: First Name, Username, Email Address, Confirm Email Address, Password, and Confirm Password. Each field is a white input box with a light gray border. At the bottom of the form is a green button with a white lock icon and the text "Sign Up". Below the button are two links: "Already have an account? Login" and "Forgot your password?".

Figure 10: Signup Page

4.1.8 User Information Page (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirement 3.2.1.7.

Description: Figure 11 allows a user or site administrator to view recent price submission history. It also allows a user to update their account password. This is done by first requesting the user's current password then asking them to create a new password and confirm it by entering it twice as shown in figure 11.

Savvy Search Submit About Hello, Taylor

User Information for Taylor

Submitted Price History

The table below is a list of the products/services this user has already submitted to Savvy.

Product Name	Date Submitted	Price Submitted
Folger's Coffee Grounds	10/6/2015	\$3.25
Dozen Eggs	10/7/2015	\$5.63
Tropicana Orange Juice	10/8/2015	\$7.64

Change Account Password

A link to reset your password will be sent to the email you signed up for Savvy with.

[Send Reset Password Instructions](#)

Home Submit About Login Sign Up Search

All Rights Reserved Savvy

Figure 11: User Information Page

4.1.9 Account Recovery Page (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirement 3.2.1.5 and 3.2.1.6.

Description: The user will visit Figure 12 when they forget their account password. The user can access the account recovery page by clicking “Forgot Password” shown on both Figures 9 and 10. The user will input their email address they used to signup with and the administrator will send them an email with instructions for creating a new password.

Savvy Search Submit About Login Register

Account Recovery

To recover your account please enter the email address you used to sign up for Savvy in the field below

Email Address

[Reset Password](#)

A link and instructions for recovering your account will be sent via email

Home Submit About Login Sign Up Search

All Rights Reserved Savvy

Figure 12: Account Recovery Page

4.1.10 Admin Dashboard (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirements 3.2.1.3 and 3.2.2.1.

Description: Figure 13 is for use by the website administrators. Here they have the options to review user information or price history. The administrative capabilities are further detailed in section 4.1.11. Each of these options are a button which will take the administrator to the respective page on the site. Figure 13 will not be visible or accessible by a non-admin user of the site. When an administrator user logs in they will be taken to this Admin Page.

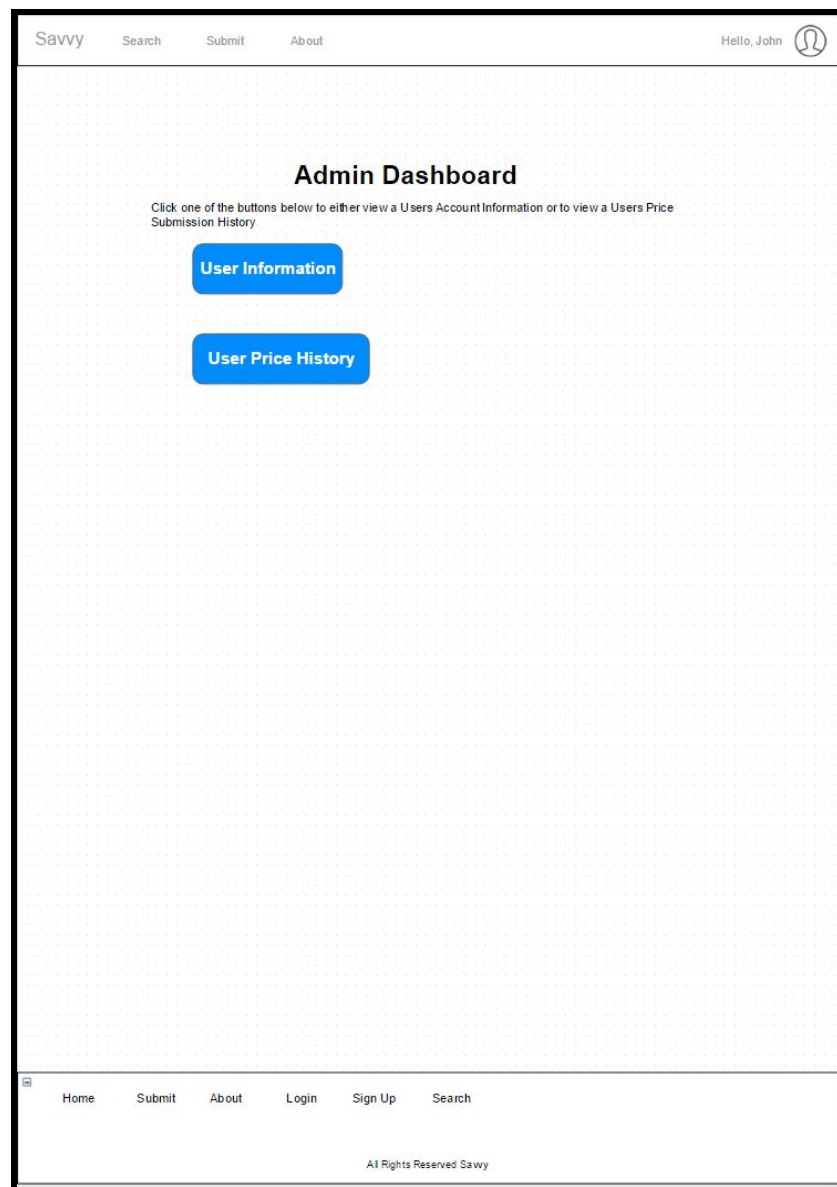


Figure 13: Admin Dashboard

4.1.11 Administrative View - List of All Users

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirements 3.2.2.1 through 3.2.2.4.

Description: Figure 14 allows the site administrators to view a table of all users. There is also the option to search for specific users by using the search box. The administrator can reset a user's password by clicking the 'Reset Password' hyperlink. The administrator can also create other administrators by using the check box next to users' information. If the administrator clicks on the hyperlinked username, Figure 15 will appear.

The screenshot displays a web interface titled "User Information". At the top, there is a navigation bar with links: Savvy, Search, Submit, About, and a user greeting "Hello, John" with a profile icon. Below the title, there is a search box with the placeholder text "Search for a User by name, id, or email" and an example "ex. John Doe". Below the search box, there are three bullet points: "Click on a username in the table below to view and edit user settings.", "Click the Reset Password link in the table below to send a user an email with instructions to reset his/her password.", and "Tick Make Admin in the table below if you wish to make a user an Administrator of the site." Below the instructions is a table with the following data:

ID	First name	Email Address	Username	Reset Password	Make Admin
12312546752	John	john@doe.com	johndoe231	Reset Password	<input type="checkbox"/>
33737373730	Taylor	taylor@test.com	jammindude92	Reset Password	<input checked="" type="checkbox"/>
86905486548	Bob	Bob@test.com	BobisTheBest1	Reset Password	<input type="checkbox"/>

At the bottom of the interface, there is a footer with links: Home, Submit, About, Login, Sign Up, Search, and a copyright notice "All Rights Reserved Savvy".

Figure 14: Administrative View - List of All Users

4.1.12 Administrative View of Specific User Page (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirements 3.2.1.6 and 3.2.2.1.

Description: Figure 15 allows administrators to view a specific user's information and reset a user's password after the administrator clicks on the user's hyperlinked username in figure 14. This is useful if the user contacts the administrators about not being able to login to the site and the administrator can then go right to the user's information page. The site displays a table of the user's price submissions to ensure that prices are accurate and also to deter abuse of the site.

The screenshot displays the 'User Information for Taylor' page. At the top, there is a navigation bar with 'Savvy', 'Search', 'Submit', and 'About' links, and a user greeting 'Hello, John' with a profile icon. The main content area is titled 'User Information for Taylor'. Below this, there is a section for 'Submitted Price History' with a note: 'The table below is a list of the products/services this user has already submitted to Savvy.' A table follows with three columns: 'Product Name', 'Date Submitted', and 'Price Submitted'. The table contains three rows of data. Below the table, there is a 'Change Account Password' section with a message: 'A link to reset your password will be sent to the email you signed up for Savvy with.' and a blue button labeled 'Send Reset Password Instructions'. At the bottom, there is a footer with links: 'Home', 'Submit', 'About', 'Login', 'Sign Up', and 'Search', and a copyright notice: 'All Rights Reserved Savvy'.

Product Name	Date Submitted	Price Submitted
Folger's Coffee Grounds	10/6/20 15	\$3.25
Dozen Eggs	10/7/20 15	\$5.63
Tropicana Orange Juice	10/8/20 15	\$7.64

Figure 15: Administrative View of Specific User Page

4.1.13 Receipt Uploader (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirements 3.2.3.3 and 3.2.3.4.

Description: Figure 16 will allow users to upload a photo of a receipt and verify product and price information. After the user uploads a receipt the form at the bottom of the page is populated with items and pricing information that we were able to identify on the receipt. The user could verify the information here and make any changes, if necessary. Figure 16 will also display a preview of the image that the user uploaded.

The screenshot shows a web application titled "Savvy" with a navigation bar containing "Search", "Submit", and "About". A user is logged in as "Taylor". The main heading is "Receipt Uploader". Below this, there is a section for uploading a receipt image, with a text input field, a "Browse" button, and an "Upload" button. A message states "The receipt image should look like this:" followed by a placeholder for a sample receipt. Another message states "Here is your receipt:" followed by a placeholder for the user's uploaded receipt. Below these, a table displays the extracted product and price information, which the user is asked to verify and correct if necessary. The table lists three items: Folger's Coffee Grounds (\$8.49), Dozen Eggs (\$3.00), and Kool Aid Fruit Punch (\$2.99). A "Submit Receipt" button is located below the table. A green success message at the bottom states "Well done! You successfully added a price for a product!". The footer contains links for "Home", "Submit", "About", "Login", "Sign Up", and "Search", along with the text "All Rights Reserved Savvy".

Product Name	Product Price
Folger's Coffee Grounds	\$8.49
Dozen Eggs	\$3.00
Kool Aid Fruit Punch	\$2.99

Figure 16: Receipt Uploader

4.1.14 FAQ Page (WA, CS)

Type: User Interface Screen

Requirement: This entity fulfills or partially fulfills requirements 3.2.3.1 and 3.2.3.6.

Description: The FAQ page will be accessible through the navigation bar. The FAQ page will be a resource for users who have questions in regards to site usage.

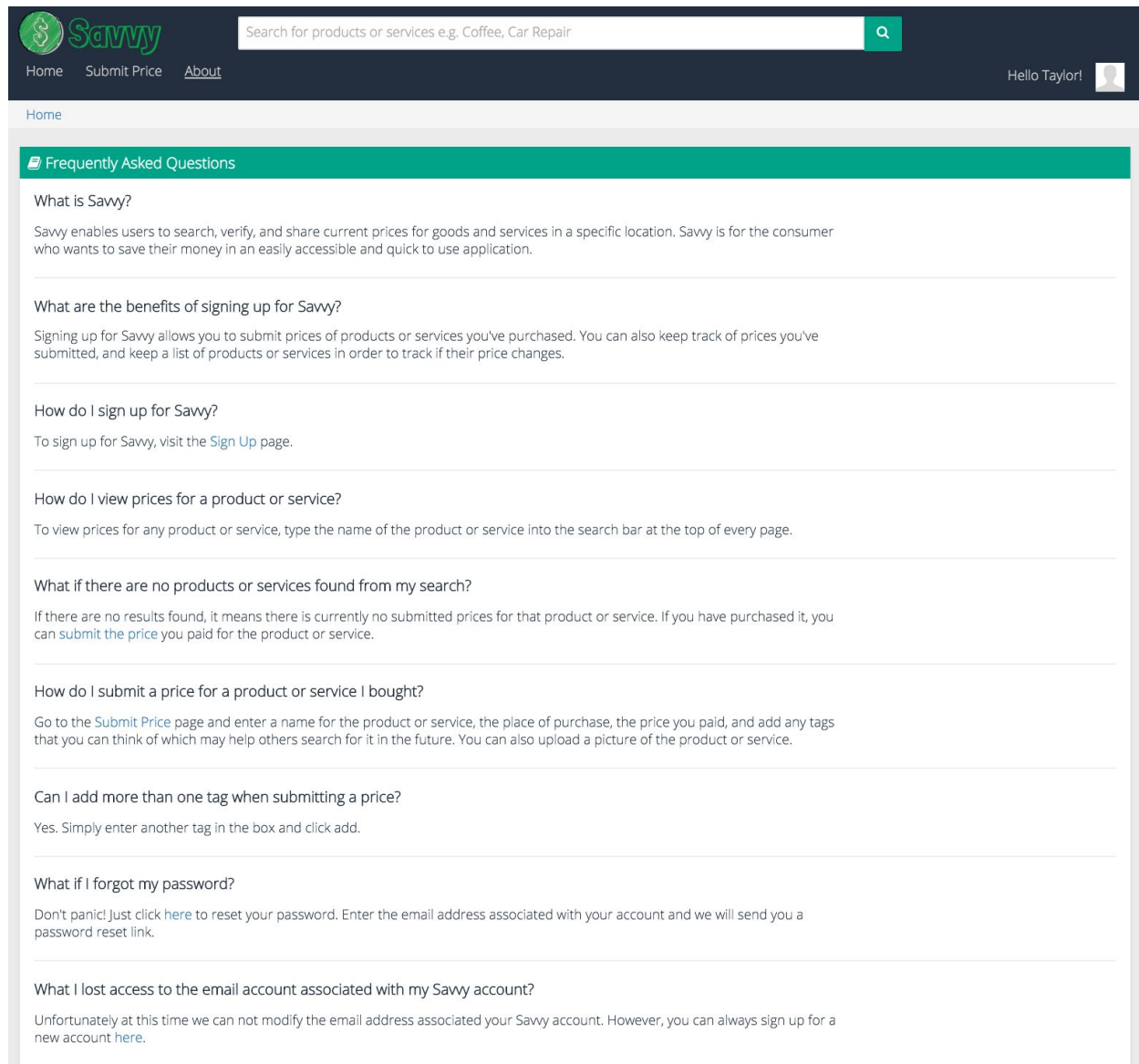


Figure 17: FAQ Page

4.2 Data Schema

4.2.1 Data Flow Diagrams (CM)

As shown in figures 18 and 19 the data flow diagrams depict the sequence of data flow throughout the system. There are 9 processes represented in the diagram and they are denoted as P1 through P9. The processes are more further explained in section 4.2.2. This diagram depicts the functional requirements 3.2.1 through 3.2.7. This diagram does not contain an administrator because all users of Savvy have the same access to every process. It is the processes themselves that conduct a privilege check.

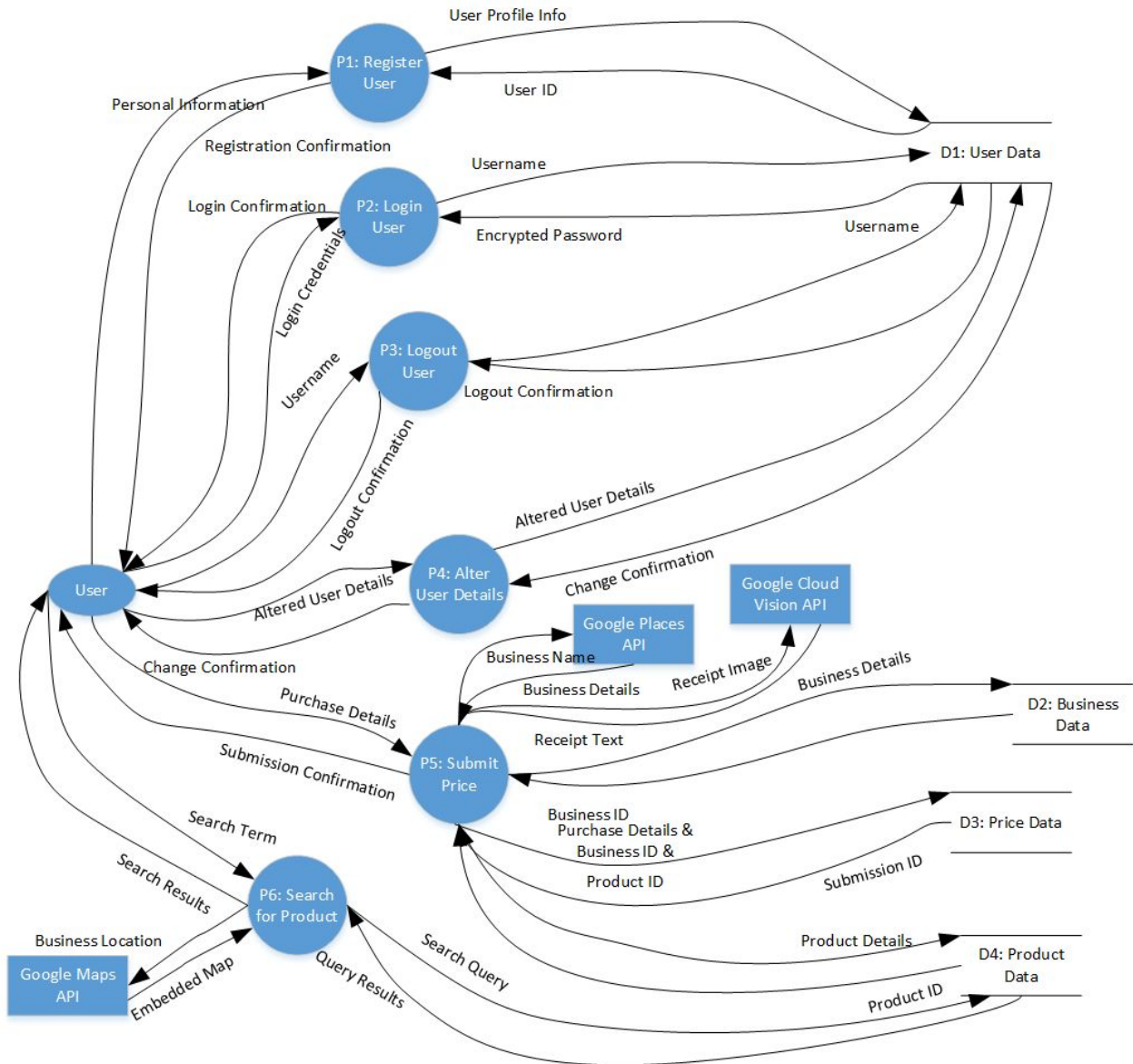


Figure 18: Data Flow Diagram 1.1

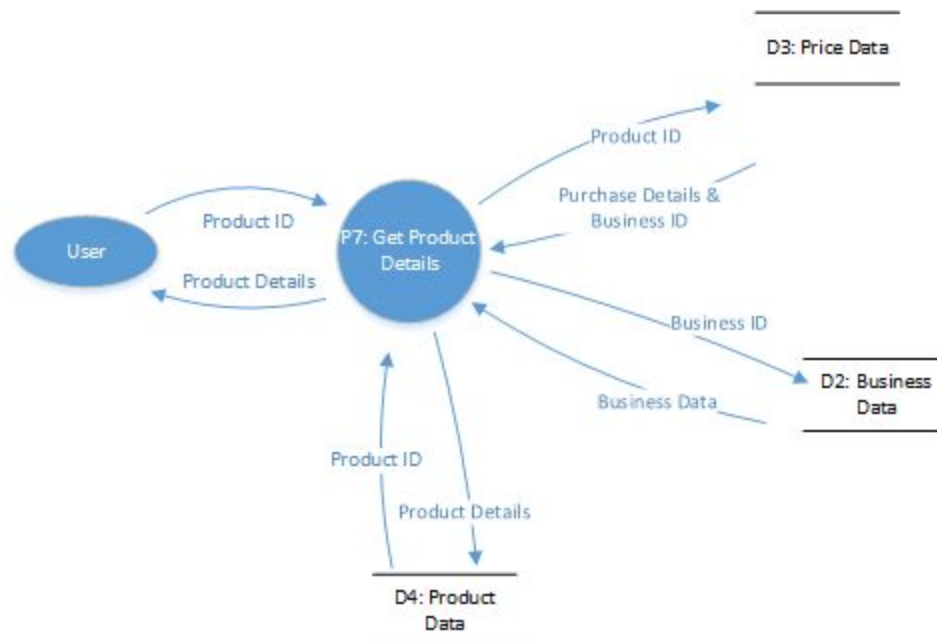


Figure 19: Data Flow Diagram 1.2

4.2.2 Design Entities (CM)

4.2.2.1 P1: Register User

The Register User process allows a user to register a user account with the system. The registration process allows for regular users to be created; however, administrative roles are granted to users by other administrators. The information provided by the user is saved in the user entity, which is detailed in section 4.3.1.

4.2.2.2 P2: Login User

The User Login process allows a user to login to the system which will provide them access to areas of the system that require authentication. The login process uses the information provided in the registration.

4.2.2.3 P3: User Logout

The User Logout process allows a user to log their user account out of the system. When the user logs out of the system the information associated with their account will be saved.

4.2.2.4 P4: Alter User Details

The Alter User Details process allows a user to change the details of their user account. The user details that can be changed are the same information is inputted during the registration process. This process include the abilities to reset password and change first name. The information is kept in the user entity which is detailed in section 4.3.1.

4.2.2.5 P5: Submit Price

The Submit Price process allows a user to submit details regarding a purchase they made and the price they paid. The details are saved in the business, product, and price submission entities which is detailed in sections 4.3.2 through 4.3.4.

4.2.2.6 P6: Search Products

The Search Products process allows a user to search the available product names for a product that matches their search terms. The information is pulled from the business, product, and price submission entities which are further detailed in sections 4.3.2 through 4.3.3.

4.2.2.7 P7: Get Product Details

The Get Product Details Process allows a user to view the details of a selected product. The information is pulled from the business, product, and price submission entities which are further detailed in sections 4.3.2 through 4.3.3.

4.2.3 Entity Relationship Diagram (CM)

Because Savvy is not entirely object oriented, an ERD has been created seen below in Figure 20. The ERD has four entities related to the logical database structure in sections 3.4.1 to 3.4.3 in the SRS.

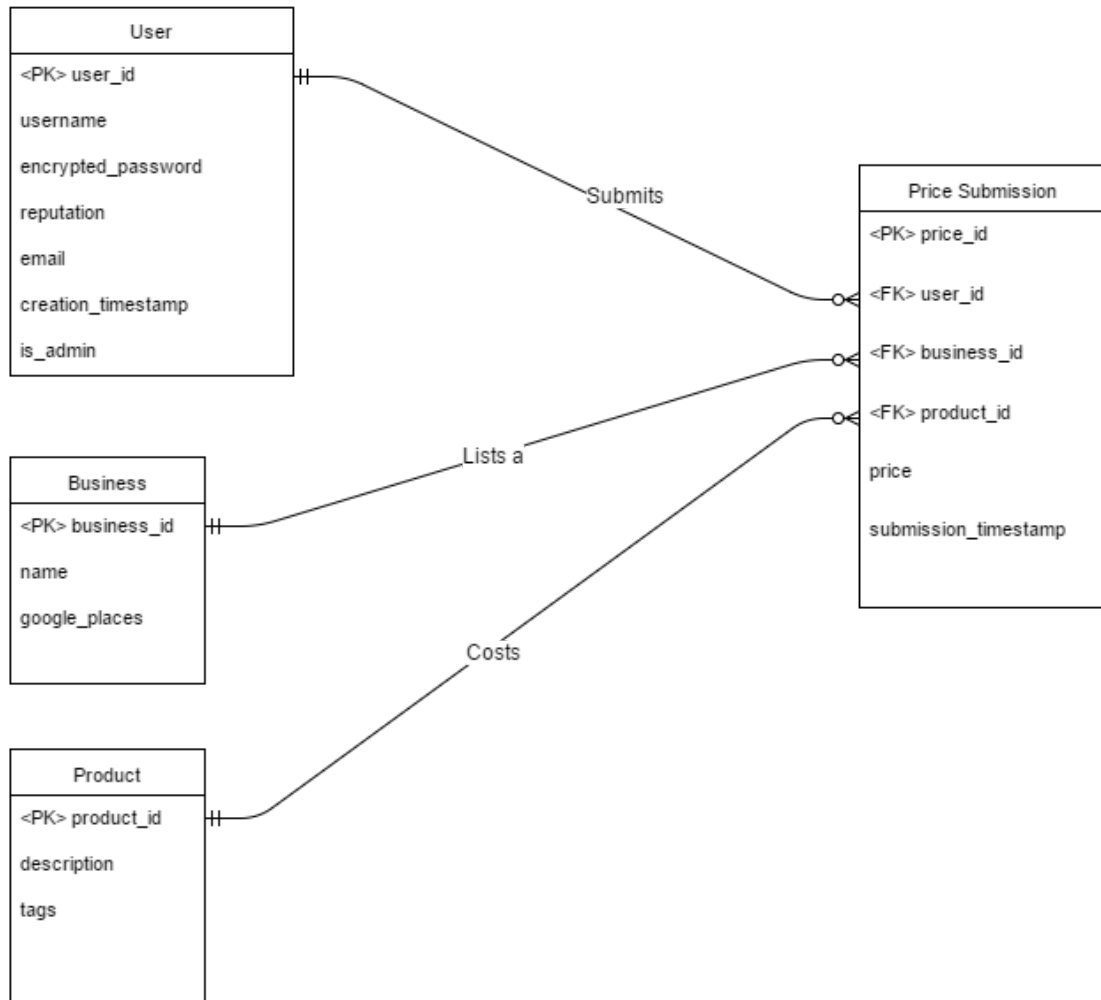


Figure 20: Entity Relationship Diagram

4.3 Data Entities

4.3.1 User (RT)

Type: Entity

Requirement: This entity fulfills requirements 3.2.1.1 through 3.2.2.4.

Description:

As shown in Figure 21, the User entity holds user information and is related to Figures 18 & 19 Table D1 “Users Data”. It contains the following attributes:

Key	Name	Type	Required?	Comment
PK	user_id	String	Yes	System generated string which uniquely identifies each user.
	username	String	Yes	The user-created username.
	encrypted_password	String	Yes	The user’s password after being encrypted.
	reputation	Integer	No	A score based on price feedback from other users.
	email	String	Yes	The user’s email address.
	creation_timestamp	Timestamp	Yes	The date the user account was created, system generated.
	is_admin	Boolean	Yes	A value which determines if a user is an admin user. It is set to 1 if the user is an admin, the default value is 0.

Figure 21: User Data Entity Description

4.3.2 Business (RT)

Type: Entity

Requirement: This entity fulfills requirements 3.2.3.1 and 3.2.4.3.

Description:

As shown in Figure 22, the Business entity holds information about businesses and is related to Figures 18 & 19 Table D2 “Business Data”. It contains the following attributes:

Key	Name	Type	Required?	Comment
PK	business_id	String	Yes	A system generated string which uniquely identifies each business.
	name	String	Yes	The name of the business.
	google_places	String	Yes	Information gathered via the Google Places API.

Figure 22: Business Data Entity Description

4.3.3 Product (RT)

Type: Entity

Requirement: This entity fulfills or partially fulfills requirements 3.2.3.5 through 3.2.3.7, and 3.2.4.1.

Description:

As shown in Figure 23, the Product entity holds information about products and services offered by businesses and is related to Figures 18 & 19 Table D4 “Products Data”. It contains the following attributes:

Key	Name	Type	Required?	Comment
PK	product_id	String	Yes	A system generated string which uniquely identifies each product or service.
	description	String	Yes	A user-entered description of the product or service.
	tags	String	No	See section 1.2 for reference.

Figure 23: Product Data Entity Description

4.3.4 Price Submission (RT)

Type: Entity

Requirement: This entity fulfills or partially fulfills requirements 3.2.1.2, 3.2.2.2, 3.2.3.1, 3.2.3.2, 3.2.3.8, and 3.2.5.2.

Description:

As shown in Figure 24, the Price entity holds information about each price that is submitted by a user and is related to Figures 18 & 19 Table D3 “Price Data”. It contains the following attributes:

Key	Name	Type	Required?	Comment
PK	price_id	String	Yes	A system generated string which uniquely identifies each price submission.
FK	user_id	String	Yes	The user_id of the user who submitted the price.
FK	business_id	String	Yes	The business_id of the business at which the product or service was purchased.
FK	product_id	String	Yes	The product_id of the product or service that was submitted.
	price	Integer	Yes	The user-entered price paid for the product or service.
	submission_timestamp	Timestamp	Yes	The date the price was submitted, system generated.

Figure 24: User Data Entity Description