**BratSpot**

**Software Requirements Specification**

**Version #2**

**Team 12**

**Project Manager:** Michelle Moore

**Mentor:** Tolu Idowu

**Team Members:** Nathaniel Behymer, Daniel Hixon, Daniel Meserve, Nathan Stewart

**Revisions**

| **Version** | **Primary**  **Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
| 1 | Nathaniel Behymer, Daniel Hixon, Daniel Meserve, Michelle Moore, Nathan Stewart | Details project objectives, scope, features, use cases, user stories, project management plan, and requirements | 10/27/21 |
| 2 | Nathaniel Behymer, Daniel Hixon, Daniel Meserve, Michelle Moore, Nathan Stewart | Includes high-level design | 11/10/21 |

**Review History**

| **Reviewer** | **Version Reviewed** | **Date** |
| --- | --- | --- |
| Tolu Idowu | 1 | 10/26/21 |
| Tolu Idowu | 2 | 11/10/21 |

**Table of Contents (version 2)**

1. Introduction
   1. Purpose
   2. Project Objective
   3. Project Scope
   4. Definitions, Acronyms, and Abbreviations
   5. Overview
2. General Description
   1. Project Features / Functions
   2. User Stories
   3. Use Cases
3. Team Collaboration and Documentation Tools
4. Project Management Plan
5. Requirements Specification
   1. Business Requirements
   2. User Requirements
   3. Functional Requirements
   4. Non-Functional Requirements
6. High-level Design
   1. Security
   2. Hardware
   3. User Interface
   4. Architecture
   5. Database
   6. Top-Level Classes
   7. Data Flows and States
   8. Reports
7. **Introduction**

This is a software requirements specification, or SRS, document for a point-of-sale application created for hot dog vendors, called BratSpot (pronounced “brotspot”). This document will specify the purpose, overall description, and requirements of BratSpot. It will serve as a reference for designers, developers, and testers to ensure that the project requirements are fulfilled. Project management may use the requirements to estimate resources and development timelines, as well as to monitor progress against the plan.

* 1. **Purpose**

The purpose of this project is to improve hot dog vendors’ productivity by enabling them to do more than input an order for a generic hot dog and calculate payment. With BratSpot, hot dog vendors will be able to input customized orders, track inventory, and display filtered search results. By simplifying inventory management processes and offering customization, hot dog vendors can differentiate themselves from others to build brand loyalty in an ever-evolving mobile food vending landscape.

* 1. **Project Objective**

Add 3 features to increase the functionality of a point-of-sale application for hot dog vendors by May 10, 2022.

* 1. **Project Scope**

BratSpot will be a mobile point-of-sale application that hot dog vendors can use to conduct their business. In addition to basic functions, such as inputting an order and calculating payment, the application will allow users to:

* + Offer customers more customization options for hot dogs
  + Keep track of inventory
  + Receive alerts when stock is running low
  + Automatically reorder items when stock is low
  + View filtered menu results

Users will have access to the application anywhere where they can access a Wi-Fi network or cellular data.

* 1. **Definitions, Acronyms, and Abbreviations**
  + **MOSCOW**: Prioritization technique to rate the importance of the delivery of each requirement - Must have, Should have, Could have, Won’t have
  + **POS**: Point of Sale
  + **SRS**: Software Requirements Specification
  1. **Overview**

The requirements will be detailed in the following sections:

* **Section 2**: Project features/functions, use cases, and user stories
* **Section 3**: Collaboration and documentation tools
* **Section 4**: Project Management Plan
* **Section 5**: Business requirements
* **Section 6**: User requirements
* **Section 7**: Functional requirements
* **Section 8**: Non-functional requirements

1. **General Description**
   1. **Project Features / Functions**

The 3 features that will be added to BratSpot:

* + **Hot dog customization**: vendors will be able to add and remove ingredients to customize hot dog orders
  + **Inventory management system**: vendors will be alerted when ingredient quantities drop below a certain amount and uses automation to order more stock.
  + **Menu filter**: vendors will be able to view filtered menu options based on selected ingredients to minimize the risk of allergic reaction and for ease of ordering.
  1. **User Stories**
     1. Create an interface for vendors to edit and keep track of orders
     2. Create inventory tracking system
     3. Create automation that will re-order items after they fall below x amount
     4. Add menu filtering capability
  2. **Use Case**
     1. Given a customer's hot dog order, when the vendor inputs the order with the chosen toppings, the order with the correct toppings should appear on the ticket.
     2. Given an ingredient that falls below x amount, when the quantity of stock goes below a specified threshold, the vendor will be alerted and the items will be automatically ordered.
     3. Given a menu search function, when a customer asks which hot dog contains x ingredient, the vendor should be able to display the filtered menu.

1. **Team Collaboration and Documentation Tools**

All team communication takes place on Microsoft Teams. GitHub is the version control system used for this project. Google Drive was used to collaborate on the PowerPoint that will be used for our project pitch presentation.

1. **Project Management Plan**

Graphical user interface, application

Description automatically generated

Timeline

Description automatically generated

1. **Requirements Specification**
   1. **Business Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| BR1 | We should collect user feedback to help give insights on possible improvements | S |
| BR2 | We need to be able to help customers on an individual basis if problems arise | M |
| BR3 | BratSpot should be available for anyone to use | S |

* 1. **User Requirements**

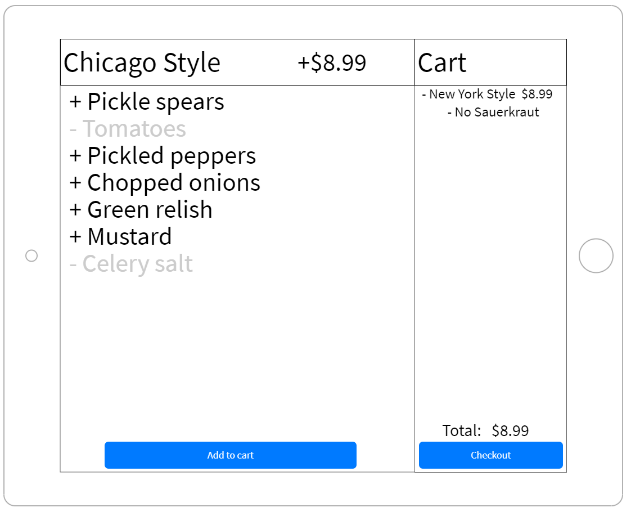
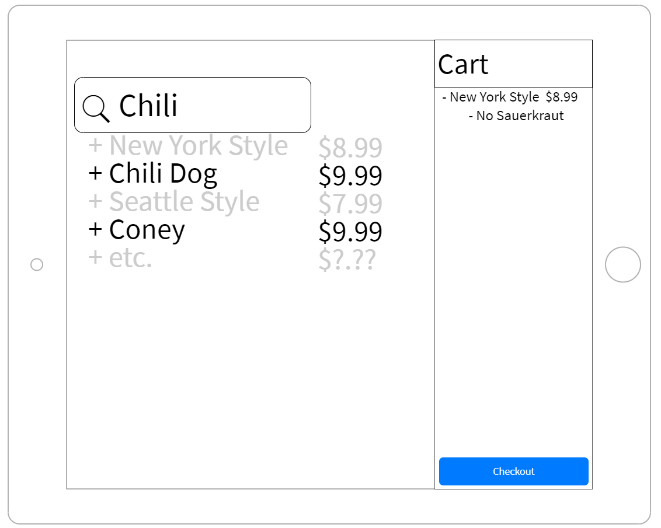
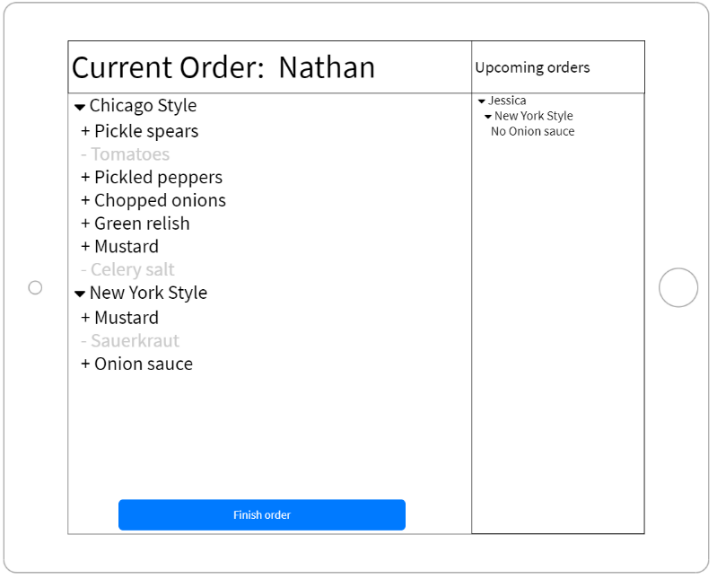
| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| UR1 | BratSpot will allow customers to input customized orders through a menu. | M |
| UR2 | BratSpot will take customized ingredients and allergy information into account to filter out redundancy for the end user and improve the ordering process. | S |
| UR3 | BratSpot’s menu will display quantities of items on the vendor side with alerts on low items, so that regardless of the automatic ordering vendors will be apprised of low stock. | M |

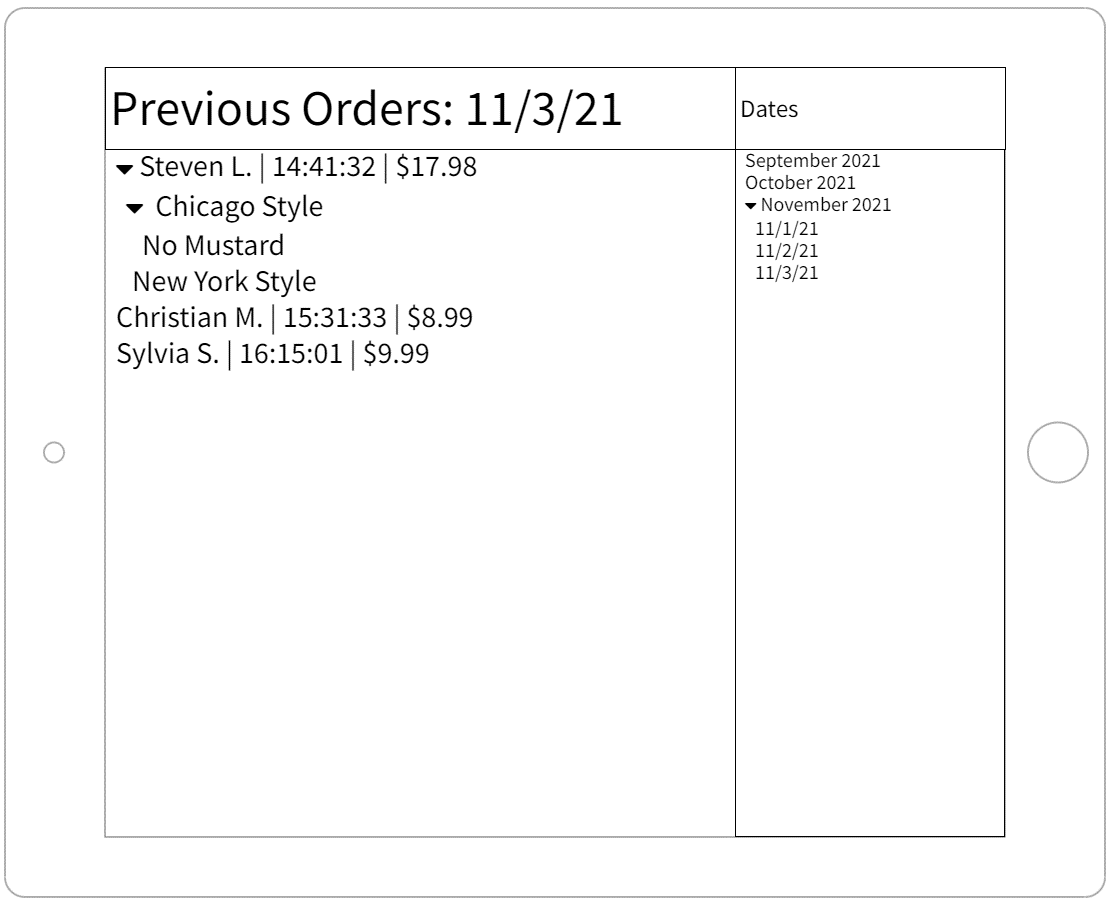
* 1. **Functional Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| FR1 | When quantities of stocked items reach a low threshold, BratSpot should automatically order more | S |
| FR2 | Once new items have arrived, BratSpot will add them to the inventory database | S |
| FR3 | When a customer makes an order, BratSpot will send the custom order to the vendor | M |
| FR4 | When a customer makes an order, BratSpot will store the data in a database for later viewing | C |
| FR5 | As a customer searches for specific ingredients or toppings, the system will filter to ones that match | M |

* 1. **Non-Functional Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| NFR1 | BratSpot will either email or text the vendor a verification code to prove their identity. | M |
| NFR2 | BratSpot will require username and password to create an account. | M |
| NFR3 | BratSpot should not allow login unless correct username and password are provided. | M |
| NFR4 | The BratSpot app should have bug fixes and maintenance updates regularly. | M |
| NFR5 | BratSpot should be able to be used on multiple mobile platforms like Apple, Android, etc. | S |
| NFR6 | BratSpot shouldn’t use a lot of storage on mobile devices. | S |
| NFR7 | When users put in their orders, the order will be updated and displayed within seconds. | M |

1. **High-Level Design**
   1. **Security**
      1. **Data** – As the application will have user logins and passwords, the data should be encrypted. The data about which products are selling should also be encrypted for market security. Therefore, the application will encrypt data using the bcrypt module, which will enable SHA-256 encryption for the passwords and files.
      2. **Network** – Because of the aforementioned data vulnerability, the network will also have security measures in place to prevent data theft or loss. The application and its server will use HTTPS to secure traffic.
   2. **Hardware Requirements**
      1. For servers, we will implement a cloud-based server. This will allow the user to access the data they need and update it whenever and wherever they need. Offloading the server to a cloud-based provider also helps the vendor keep their focus elsewhere.
      2. BratSpot will use wi-fi and cellular data to keep connection to the cloud-based server mentioned above. This will allow for a connection when in an area with the vendor’s own network they already have set up, or even on the go with cellular data for hot dog trucks, for example.
      3. To allow users to order at the point of sale, BratSpot will use touchscreen devices like tablets and smart phones. This will make the sale much easier and require less hardware while allowing for less mistakes in the ordering phase.
   3. **User Interface Requirements** 
      1. Navigating the app will be through the use of the touch screen devices the app will run on. Screens will have back buttons to go back a menu and redo options, and going forward will be through large confirmation buttons that tell you exactly what they do (ex. Add order to cart)
      2. For ordering, there will be three main menus. The first is a search menu, where you can select what you want from the menu by filter through names and ingredients in hot dogs. The second menu is where you can add and remove topping and condiments from selected hot dogs, and the third is a checkout menu where you can pay and add your name to the order. For the vendor, the main menu will be a list of orders with the customer’s name and the toppings for each hot dog on the order, as well as a list of upcoming orders.
      3. A rough wireframe of the 2 main menus for the customers (excluding checkout screen, which would just be name and payment info with a summary of what is ordered)
      4. A wireframe sketch of the vendors menu where they can view their orders, and a page where they can view orders (and additional information if needed) from the database.



* 1. **Architecture**
     1. For this application, we will use a 2-tier client-server architecture (thin client fat server). The frontend will be stored on the client machine and will utilize touch screens for navigation. The business and data logic will be handled on the server to offload the workload and reduce upfront costs. We will use PostgreSQL as our database because it is highly reliable, scalable, stable, and secure. It is also open-source, which will further reduce upfront costs as there are no licensing fees.
  2. **Database**
     1. As previously stated, we will use PostgreSQL as our database to store user account information, purchases, and inventory information so we can gather statistics regarding customer preferences and inventory. Gathering primary data will help vendors recognize what products are popular with customers, and how much stock will be needed to ensure they don’t run out prematurely. Vendors should not be able to directly access this database, only their own data.
  3. **Top-Level Classes**Diagram

     Description automatically generated
  4. **Data Flows and States**

Diagram

Description automatically generated

* 1. **Reports**

6.8.1. The application will create customer reports, as one of our main features has been to provide vendors with a view of popular items/combinations so that the vendor can strategize their business plan around it.