

## Release Planning Document: Event Management and Ticketing Platform

### Ticket Express

#### Team Members:

- Sinatra Almeida (7060049)
- Jacob Applebaum (7215031)
- Joycelyn Chan (7239486)
- Taylor Chapman (7178577)
- Parker TenBroeck (7376726)
- Benjamin Williams (6953954)

#### 1. Introduction

This document describes the preliminary release planning for *Ticket Express*, a software-as-a-service tool for event management and ticketing, as well as a term project for the course COSC 4P02 at Brock University. *Ticket Express* is a state-of-the-art product designed for professionals in the entertainment industry and common consumers. The purpose of this document is to record the process used to construct a detailed release schedule. Specifically, the document describes the collected user stories, product backlog, and sprint backlogs.

#### 2. User Stories

**Priorities:** High, Medium, Low

**Kind:** Feature, Bug, Security, Enhancement, Optimization

**Roll:** Organizer, Attendee, System Admin

##### User Story 1: (Organizer) Medium

As an organizer, I want to quickly verify that an attendee has a valid ticket for an event so that the check-in process is smooth.

##### **Tasks:**

- Setup a system for QR code generation
- Setup a system for scanning QR codes and retrieving the value
- Expose an API endpoint to verify a specific ticket is a valid ticket for a specified event

##### User Story 2: (Attendee) High

As an attendee, I want to browse, search, and filter events by date or popularity, including both past events for archival purposes and future events I might want to attend.

**Tasks:**

- <select> form element with options for date and popularity (# of people signed up)
- Construct the database in a way to allow counting the number of people signed up for an event and have a popularity metric.
- Expose a search endpoint that accepts several parameters for sorting and filtering.
- Generate SQL depending on the filter and sorting parameters present in search requests.

**User Story 3: (Organizer)** Low

As an organizer, I want an application that can handle event organization and ticketing for over 1000 people, and I also want to be able to view data and analytics for individual events and aggregate performance data from multiple events to assess the success of my concerts.

**Tasks:**

- Make separate table in DB for keeping track of event registrations
- Analytics front end will send API commands then visualize the data for user
  - Graphs, Pie charts

**User Story 4: (Attendee)** High

As an attendee, I want to create a personalized account with a custom profile.

**Tasks:**

- Login page
- Sign up page
- Make dedicated table in DB for accounts

**User Story 5: (Organizer/Attendee)** Medium

As an organizer/attendee, I want to ensure my payments are secure and have access to a wide variety of payment methods.

**Tasks:**

- Do research on monetary transaction systems
- Implement a secure solution in the event registration flow before you get your ticket

**User Story 6: (Organizer)** High

As an organizer, I want to create and manage multiple events and customize each event to match the attendee bases who it will be marketed to.

**Tasks:**

- Create API for creating event drafts and publishing finished event drafts to the organizers profile.
- Event Customization: create templates for different event types, enable customization (logos, themes, etc.)
- Event Management: Design a dashboard to list and manage events, add features to edit event details, monitor attendee registration and ticket sales.
- Notifications and Reminders: implement auto email notifications for event creation and event reminders/updates for attendees.

#### **User Story 7: (Organizer/Attendee/Admin) High**

As an organizer/attendee/admin, I want to log in to the platform, so that I can access my personalized dashboard and perform my role-specific actions.

##### **Tasks:**

- Design the login interface
- Set up authentication system
- Role-Based Access: Set up access control
- “Forgot Password” Workflow: implement backend logic for email verification and password updates.
- Test login functionality

#### **User Story 8: (Attendee) Low**

As an attendee, I would like an option of resending an email of my tickets once I have purchased them.

##### **Tasks:**

- Implement an email notification system
- Implement a feature that sends the generated QR code to the email address of an account upon the user

#### **User Story 9: (Attendee/Organizer/Admin) Medium**

As an organizer/attendee/admin, I would like to use this service effectively using the web browser on my phone.

##### **Tasks:**

- Implement responsive web design using CSS media queries.
- Implement collapsible and easy to use navbar.
- Optimize interactive elements for touch input.

#### **User Story 10: (Organizer /Attendee) Low**

As an organizer/attendee, I would like to have a calendar view of my upcoming and registered events so that I can easily track and manage the events I am organizing/attending.

**Tasks:**

- Design the calendar interface.
- Add clickable events that open detailed information.
- Test for responsiveness and usability.

**User Story Table:**

ID	Title	Description	Acceptance Criteria	Priority
1	Verify Ticket Validity	As an organizer, I want to quickly verify that an attendee has a valid ticket for an event so that the check-in process is smooth.	Attendees can display a printed/digital ticket with the QR code. Organizers can scan a QR code to verify validity.	Medium
2	Browse and Filter Events	As an attendee, I want to browse, search, and filter events by date or popularity, including both past events for archival purposes and future events I might want to attend.	Attendees can search for events by keywords.  Attendees can filter/sort events based on popularity and date.	High
3	Scalable Event Management and Analytics	As an organizer, I want an application that can handle event organization and ticketing for over 1000 people, and I also want to be able to view data and analytics for individual events and aggregate performance data from multiple events to assess the success of my concerts.	Platform that supports event creation and ticketing for over 1000 people.  Analytics dashboard should be able to display performance metrics for single/multiple events.	Low
4	Create Personalized Account	As an attendee, I want to create a personalized account with a custom profile.	Attendees can register with custom profiles.	High
5	Secure and Flexible Payments	As an organizer/attendee, I want to ensure my payments are secure and have access to	Payment methods should include credit/debit, bank	Medium

		a wide variety of payment methods.	<p>transfers, and digital wallets (e.g. Google Pay, PayPal).</p> <p>Attendees must receive a confirmation email for successful transactions.</p> <p>Payment details should be stored securely.</p>	
6	Create and Manage Events	As an organizer, I want to create and manage multiple events and customize each event to match the attendee bases who it will be marketed to.	Organizers can create, edit, update and delete event details (e.g. name, date, time, location).	High
7	User Login and Dashboard	As an organizer/attendee/admin, I want to log in to the platform, so that I can access my personalized dashboard and perform my role-specific actions.	<p>Users must be able to login with valid credentials.</p> <p>Users must be directed to their personalized dashboard based on their role.</p> <p>Restrict access for multiple invalid attempts and display an error message.</p>	High
8	Option to Resend Ticket Email	As an attendee, I would like an option of resending an email of my tickets once I have purchased them.	<p>A resend option must be available on the attendee's dashboard.</p> <p>The user receives a copy of the original ticket(s) in their email.</p>	Low
9	Mobile-friendly User Experience	As an organizer/attendee/admin, I would like to use this service effectively using the web browser on my phone.	<p>All functionalities of the software must be responsive and fully functional on mobile browsers.</p> <p>Navigation and core features must be accessible and optimized for touch.</p>	Medium

10	Calendar View for Upcoming/Registered Events	As an organizer/attendee, I would like to have a calendar view of my upcoming and registered events so that I can easily track and manage the events I am attending/organizing.	A calendar widget is displayed when registering for events, and events are stored on this calendar.  The calendar should be accessible from the attendee's dashboard.	Low
----	--	---	---	-----

### 3. Product Backlog

A prioritized list of tasks derived from the user stories in the previous section are aggregated into the following table to represent a product backlog. Each backlog entry includes the description of a specific task, an associated user story, and a priority ranking.

Task	User Story	Priority
Setup a system for QR code generation	1	Medium
Implement a system for scanning QR codes	1	Medium
Expose an API endpoint to verify a specific ticket is a valid ticket for a specified event	1	Medium
Create a webpage for the service	2	High
Construct the database in a way to allow counting the number of people signed up for an event and have a popularity metric.	2	High
Expose a search endpoint that accept several parameters for sorting and filtering	2	Medium
Generate SQL depending on the filter and sorting parameters present in search requests	2	Medium
Make separate table in DB for keeping track of event registrations	3	High
Develop database schema	3	High
Create a data analytics feature	3	Low
Develop a login page	4	Medium
Sign up page	4	Medium
Make dedicated table in DB for accounts	4	Medium
Implement a secure solution in the event registration flow before you get your ticket	5	Medium
Create API for creating event drafts and publishing finished event drafts to the organizers profile.	6	Low
Develop templates for different event types, enable customization	6	Low
Design a dashboard to list and manage events, add features to edit event details, monitor attendee registration and ticket sales	6	Low
Implement auto email notifications for event creation and event reminders/updates for attendees.	6	Low
Design the login interface	7	Low
Set up authentication system	7	Medium

Set up role-based access control	7	Low
Implement backend logic for email verification and password updates	7	Low
Test login functionality	7	Low
Implement a feature that sends the generated QR code to the email address of an account upon the user	8	Medium
Implement responsive web design using CSS media queries	9	Medium
Implement collapsible and easy to use navbar	9	Low
Optimize interactive elements for touch input	9	Low

Product Backlog link:

<https://github.com/users/DerfTastic/projects/1/views/1>

Sprint Backlog link:

<https://github.com/users/DerfTastic/projects/1/views/7>

#### 4. Sprint Backlog

Sprint backlogs are subsets of the product backlog, planned for specific sprints. *Ticket Express* will be comprised of six individual sprints, each taking two weeks to complete. The projected

Provide a breakdown of tasks for each sprint (do this for all sprints, understanding that tasks for later sprints can be revised as the project evolves).

##### 4.a) Sprint 1 (January 15<sup>th</sup> – January 29<sup>th</sup>)

Task	Priority	Status	Developers
Login Authentication	High	TODO	Sinatra Almeida, Joycelyn Chan, Taylor Chapman
Login Page	High	TODO	Sinatra Almeida, Joycelyn Chan, Taylor Chapman
Sign-up Page	High	TODO	Sinatra Almeida, Joycelyn Chan, Taylor Chapman
Generate SQL for Filtering	High	TODO	Parker TenBroeck, Benjamin Williams
Expose Search API	High	TODO	Parker TenBroeck
Expose Ticket Verification API Endpoint	High	TODO	Parker TenBroeck
Create API for Event Creation and Editing	Medium	TODO	Parker TenBroeck, Benjamin Williams

Design and Implement Database Structure	High	IN PROGRESS	Benjamin Williams
Design Style Outline for Product	High	IN PROGRESS	Sinatra Almeida, Joycelyn Chan
Develop JavaScript Frontend API with Server	High	IN PROGRESS	Sinatra Almeida, Jacob Applebaum, Joycelyn Chan, Taylor Chapman
Implement Style for Frontend	High	IN PROGRESS	Sinatra Almeida, Jacob Applebaum, Joycelyn Chan, Taylor Chapman
Filtered Search Bar	High	IN PROGRESS	Sinatra Almeida, Joycelyn Chan, Taylor Chapman
Design Style Outline for Product	High	IN PROGRESS	Sinatra Almeida, Joycelyn Chan, Taylor Chapman
Mobile Device Support	Medium	IN PROGRESS	Parker TenBroeck, Benjamin Williams
Develop Uploaded Content Database Table	Low	IN PROGRESS	Benjamin Williams
Create Framework for API Route Generation	High	COMPLETE	Parker TenBroeck
Create Static File Server	High	COMPLETE	Parker TenBroeck
Optimize Static File Server	Low	COMPLETE	Parker TenBroeck

#### 4.b) Sprint 2 (January 30<sup>th</sup> – February 12<sup>th</sup>)

Task	Priority	Status	Developers
QR Code Generation	Medium	TODO	Jacob Applebaum, Taylor Chapman
QR Code Scanning	Medium	TODO	Jacob Applebaum, Taylor Chapman
Implement Responsive Web Design (CSS)	Medium	TODO	Sinatra Almeida, Jacob Applebaum, Joycelyn Chan, Taylor Chapman
Implement Collapsible Navbar	Medium	TODO	Sinatra Almeida, Jacob Applebaum, Joycelyn Chan, Taylor Chapman



Optimize Interactive Elements for Touch Input	Medium	TODO	Sinatra Almeida, Jacob Applebaum, Joycelyn Chan, Taylor Chapman
Implement Secure Payment Method	Medium	TODO	Parker TenBroeck, Benjamin Williams

#### 4.c) Sprint 3 (February 13<sup>th</sup> – February 26<sup>th</sup>)

Task	Priority	Status	Developers
Create Analytics API	Medium	TODO	Parker TenBroeck, Benjamin Williams
Research Secure Payment Options	Medium	IN PROGRESS	Parker TenBroeck
Database Testing	Medium	IN PROGRESS	Benjamin Williams
API Testing	Medium	IN PROGRESS	Taylor Chapman, Parker TenBroeck
Framework Testing	Medium	IN PROGRESS	Benjamin Williams
Uploaded Content Server Testing	Medium	IN PROGRESS	Benjamin Williams

#### 4.d) Sprint 4 (February 27<sup>th</sup> – March 12<sup>th</sup>)

Task	Priority	Status	Developers
Add Analytics Visualization	Low	TODO	Jacob Applebaum
Password Recovery	Low	TODO	Sinatra Almeida, Parker TenBroeck

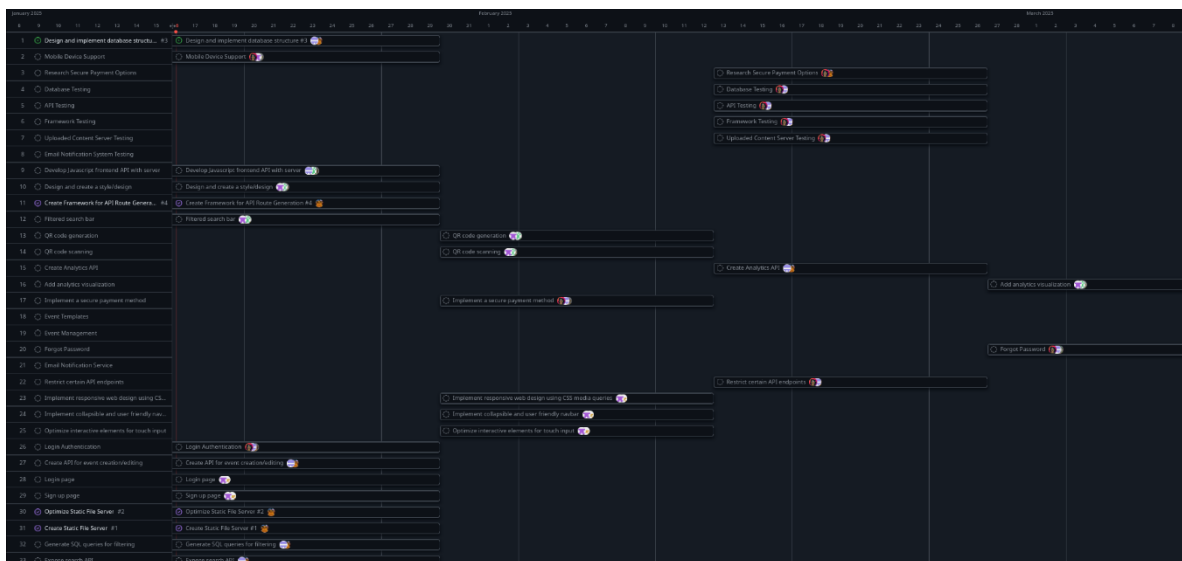
#### 4.e) Sprint 5 (March 13<sup>th</sup> – March 26<sup>th</sup>)

Task	Priority	Status	Developers
Event Templates	High	TODO	Sinatra Almeida, Jacob Applebaum, Joycelyn Chan, Taylor Chapman
Event Management	High	TODO	Jacob Applebaum, Taylor Chapman, Benjamin Williams
Email Notification Service	Medium	TODO	Sinatra Almeida, Jacob Applebaum, Joycelyn Chan, Taylor Chapman

#### 4.f) Sprint 6 (March 27<sup>th</sup> – April 9<sup>th</sup>)

Task	Priority	Status	Developers
Comprehensive Unit Testing	High	TODO	Taylor Chapman, Benjamin Williams
Refactor Code	Low	TODO	Parker TenBroeck, Benjamin Williams
Fix Identified Bugs	High	TODO	Sinatra Almeida, Jacob Applebaum, Joycelyn Chan, Taylor Chapman, Parker TenBroeck, Benjamin Williams

#### 5. Tool Screenshots



Roadmap view

The screenshot displays a 'User Stories view' in a project management application. On the left, a sidebar lists iterations: Iteration 1 (Current, 9 items), Iteration 2 (6 items), Iteration 3 (3 items), Iteration 4 (2 items), Iteration 5 (3 items), and No Iteration (0 items). The main area is a Kanban board with columns: 'has: user-story' (filter), 'Todo' (7 items, 'This item hasn't been started'), 'In Progress' (2 items, 'This is actively being worked on'), and 'Done' (0 items, 'This has been completed'). The board shows three groups of user stories. Group 1 (1 item) includes 'Expose ticket verification API endpoint' (Draft, System Admin, Feature, High, Iteration 1). Group 2 (3 items) includes 'Generate SQL queries for filtering' (Draft, Attendee, Feature, Low, High, Iteration 1), 'Expose search API' (Draft, Attendee, Feature, Low, High, Iteration 1), and 'Filtered search bar' (Draft, Attendee, Feature, Low, High, Iteration 1). Group 3 (4 items) includes 'Login page' (Draft, Attendee, Feature, Low, High, Iteration 1) and 'Sign up page' (Draft, Attendee, Feature, Medium, High, Iteration 1). Each story card has a draft icon, a title, tags for role, type, priority, and iteration, and an 'Add item' button.

User Stories view

The image shows a Jira Product Backlog board with three columns: **Todo** (20 items), **In Progress** (11 items), and **Done** (3 items). Each item is a task card with a title, priority, iteration, and assignee.

**Todo Column:**

- Login Authentication** (High, L, Iteration 1, System Admin, Security 8)
- Login page** (High, L, Iteration 1, Attendee, Feature 4)
- Sign up page** (High, M, Iteration 1, Attendee, Feature 4)
- Generate SQL queries for filtering** (High, L, Iteration 1, Attendee, Feature 2)
- Expose search API** (High, L, Iteration 1, Attendee, Feature 2)
- Expose ticket verification API endpoint** (High, M, Iteration 1, System Admin, Feature 1)
- QR code generation** (Medium, M, Iteration 2, Attendee, Enhancement 1)
- QR code scanning** (Medium, M, Iteration 2, Organizer, Enhancement 1)
- Create Analytics API** (Medium, XL, Iteration 3, Organizer, Feature 3)
- Implement responsive web design using CSS media queries** (Medium, M, Iteration 2, Enhancement 9)
- Implement collapsible and user friendly navbar** (Medium, M, Iteration 2, Enhancement 9)

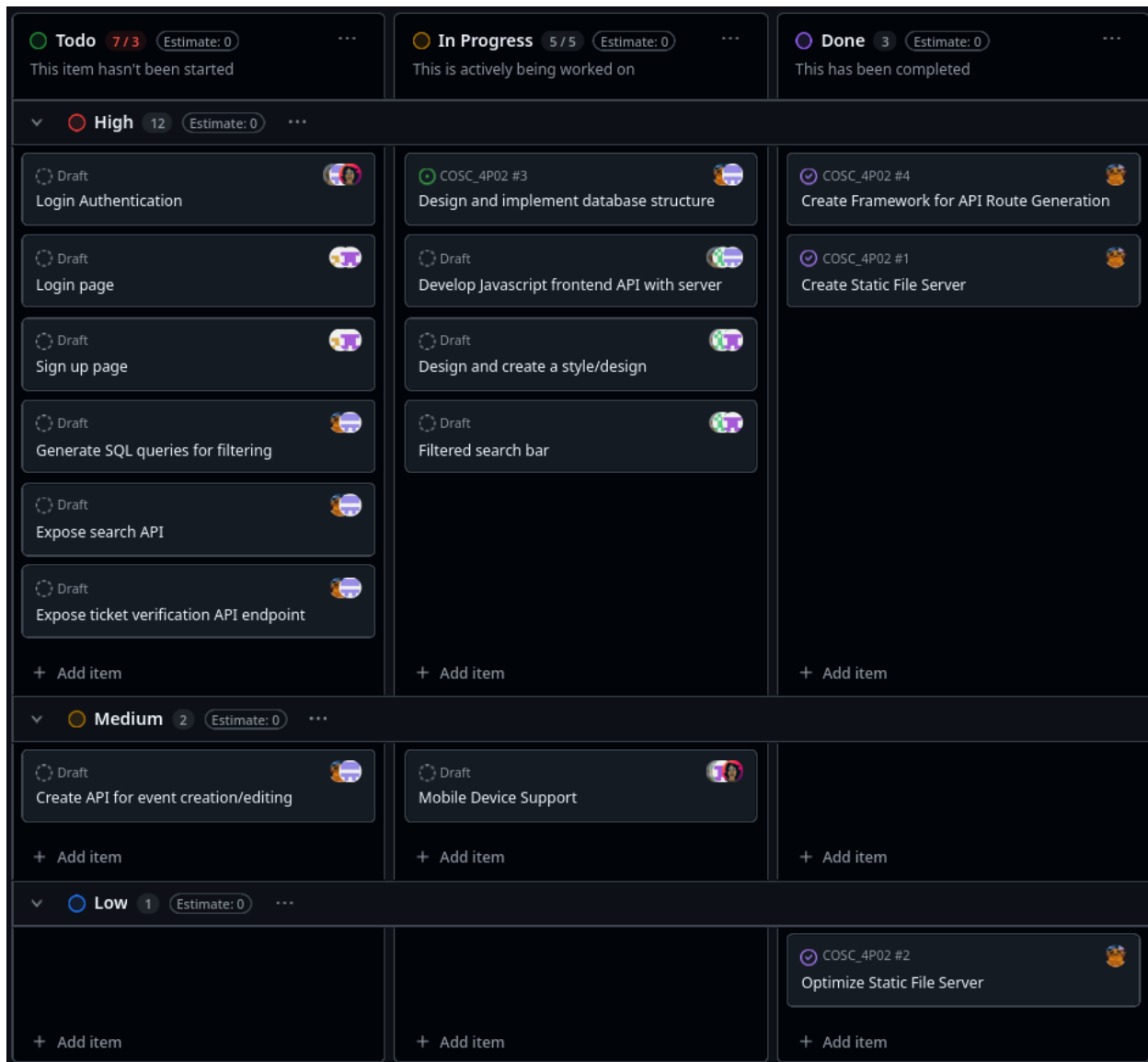
**In Progress Column:**

- COSC\_4P02 #3** (High, L, Iteration 1, System Admin, Feature)
- Design and implement database structure** (High, L, Iteration 1, System Admin, Feature)
- Develop Javascript frontend API with server** (High, XL, Iteration 1, System Admin, Feature)
- Design and create a style/design** (High, XL, Iteration 1, Feature)
- Filtered search bar** (High, L, Iteration 1, Attendee, Feature 2)
- Mobile Device Support** (Medium, XL, Iteration 1, Attendee, Feature 9)
- Research Secure Payment Options** (Medium, M, Iteration 3, System Admin, Security 5)
- Database Testing** (Medium, L, Iteration 3, System Admin, Testing)
- API Testing** (Medium, L, Iteration 3, System Admin, Testing)
- Framework Testing** (Medium, L, Iteration 3, System Admin, Testing)
- Uploaded Content Server Testing** (Medium, L, Iteration 3, System Admin, Testing)
- Email Notification System Testing** (Medium, L, Iteration 6, System Admin, Testing)

**Done Column:**

- COSC\_4P02 #4** (High, XL, Iteration 1, System Admin, Feature)
- Create Framework for API Route Generation** (High, S, Iteration 1, System Admin, Feature)
- Create Static File Server** (High, S, Iteration 1, System Admin, Feature)
- Optimize Static File Server** (Low, S, Iteration 1, System Admin, Optimization)

Product Backlog (not exhaustive)



Example Sprint

## 6. Team Contributions

Team Member	Contribution
Sinatra Almeida	Contributed to the creation of user stories, Edited and reorganized document
Jacob Applebaum	Contributed to the creation of the user stories and created GitHub repository
Joycelyn Chan	Reformatted, edited, and reorganized document

Taylor Chapman	Reviewed document
Parker TenBroeck	Contributed to the creation of user stories, sprint backlog, product backlog, and set up GitHub project
Benjamin Williams	Contributed to the creation of user stories, product backlogs, and communication with Professor/TA

## 7. Challenges and Next Steps

Organizing a coherent schedule for scrum meetings was a difficult task. At least one member of the team has a class for any given time in the week, besides the decided upon time of 3:00PM-4:00PM on Thursdays. This roadblock has prompted informal meetings with a subset of the team whenever a specific need arises. Another challenge that was presented while creating the release schedule is determining a proper sequence of events.

Moving towards the second official scrum meeting, each member of the development team will have committed some work towards the project. Namely, each entry in the first sprint backlog should be completed. Should certain tasks take longer than initially forecast, the subsequent sprint backlogs may need to be revised or refined.