CEN4010 Summer 2015
Couch Level Project Proposal

Team

Nick Petty: product owner
Caio Farias: scrum master
Avinash Rao: back-end web developer
Scott Hinden: front-end web developer

Product Name

Rendezvous

Executive Summary

Through modern communication tools, like social networks and messaging, people are more connected now than ever before. However, this connection is often purely digital, and real-world, physical interaction can be a challenge to organize and take part in. This is what Rendezvous seeks to address, we want to give people a simple tool for creating and joining events based around outdoor activities. Through the Rendezvous system, our customers will connect with each other and engage in genuine face-to-face interaction, be it by playing a sport, visiting a museum, or exploring their local parks. We are making this service available to anyone with basic computer skills, regardless of age, gender, ethnicity, gender, or religion.

In this document, certain terms may have a specific meaning in relation to the Rendezvous platform:
“Activity” refers to something people do outside, like swimming, playing baseball, or hiking.
“Event” refers to an organized occasion, at certain time and place, where people do an activity.

Competitive Analysis

Currently, a person intending to organize or join an event has several options. The following is a list of common methods to do so, and the problems with these methods.

- Social Networks (like Facebook, Twitter, Instagram)
    - Limited/excessive visibility: events may only be seen by a user’s network, or by everyone globally.
    - Not focused: events may be seen by uninterested people.
    - Connected to social status: people may avoid events after viewing the profiles of participants.
- Bulletin Boards (like Reddit and Craigslist)
    - Excessive visibility: almost all events are globally viewable.
    - Difficult to navigate: finding or creating an event requires working with a complicated user interface.
- Community groups/centers

- Very limited visibility: people must visit the center or already be part of the group.
- Inconsistent experience: each event is organized differently, so information and communication is not the same from one activity to another.

Rendezvous will address these issues by delivering two primary strengths - event visibility and user experience. Because we are service specifically focused on organizing location-aware outdoor activities, users will know that events will be seen by interested people, in the right place. Our service will provide a clean, responsive, and unified interface for finding and creating events. Customers will see a website that is intuitive, maintains a consistent design regardless of access method, and has a feature set that does exactly what they need.

User Scenarios

We expect to have three basic user interactions with our website: a new user joins the service, an existing user creates an event, and an existing user joins an event.

- New users
    1. The website will give a brief, skippable introduction to the Rendezvous service.
    2. All website features are enabled and usable, so a visitor can browse events or create their own.
    3. Once a visitor confirms creating or joining an event, they are taken to the sign-up page.
    4. After the short sign-up, their event is created or joined.
- Existing users
    1. After logging in manually or automatically, the user sees the home page.
    2. The home page has the same appearance for all users – activity list on the left pane, event list on the right pane, and navigation bar on the top.
    3. To create an event, the user selects the "create event" option from the navigation bar, and fills in necessary information in the event creation dialogue.

    -or-

    3. To join an event, the user chooses an activity from the list, and then selects an event, which opens in the full right pane. More information on the event is given, and the user has the "join" option.

Further management of events and notifications is done through the navigation bar.

Requirements and Infrastructure

Functional requirements:

1. Website – must have top navigation bar, 1/3 left navigation pane, 2/3 right data pane, dialogues, information pages. No action should require more than 5 steps.
2. User database – holds personal information and passwords, must be secure.
3. Event database – holds information about events, must be fast.

Non-functional requirements:

1. Scalability – initially host up to 1000 user accounts, with up to 100 users online concurrently. Ability to add more usage as needed.
2. Uptime – website is available 99% of the time. Servers can be down for up to 3 hours per week for maintenance, with scheduled rolling blackouts across time zones.
3. Security – personal data is held, but not financial data. High security is required to protect users' privacy.
4. Performance – no action on the website should take longer than 5 seconds to process. All functions look and work the same regardless of access method.

Infrastructure:

1. Hosting: initially on FAU's LAMP server.
2. Development tools: Notepad++, WinSCP, PuTTY, GitHub, Someone, Balsamiq.
3. Frameworks: Bootstrap for HTML, CSS, JS.
4. Database: MySQL and PHP.
5. Access methods: desktop and mobile versions of Chrome, Firefox, Safari, Internet Explorer. Desktop and mobile app.

Schedule

Sprint 1: repositories created, website front-end deployed.
Sprint 2: user database deployed, user creation feature implemented.
Sprint 3: event database deployed, event management implemented.