Gimball3000

Bluetooth Bicycle Tracker

Trillium Health

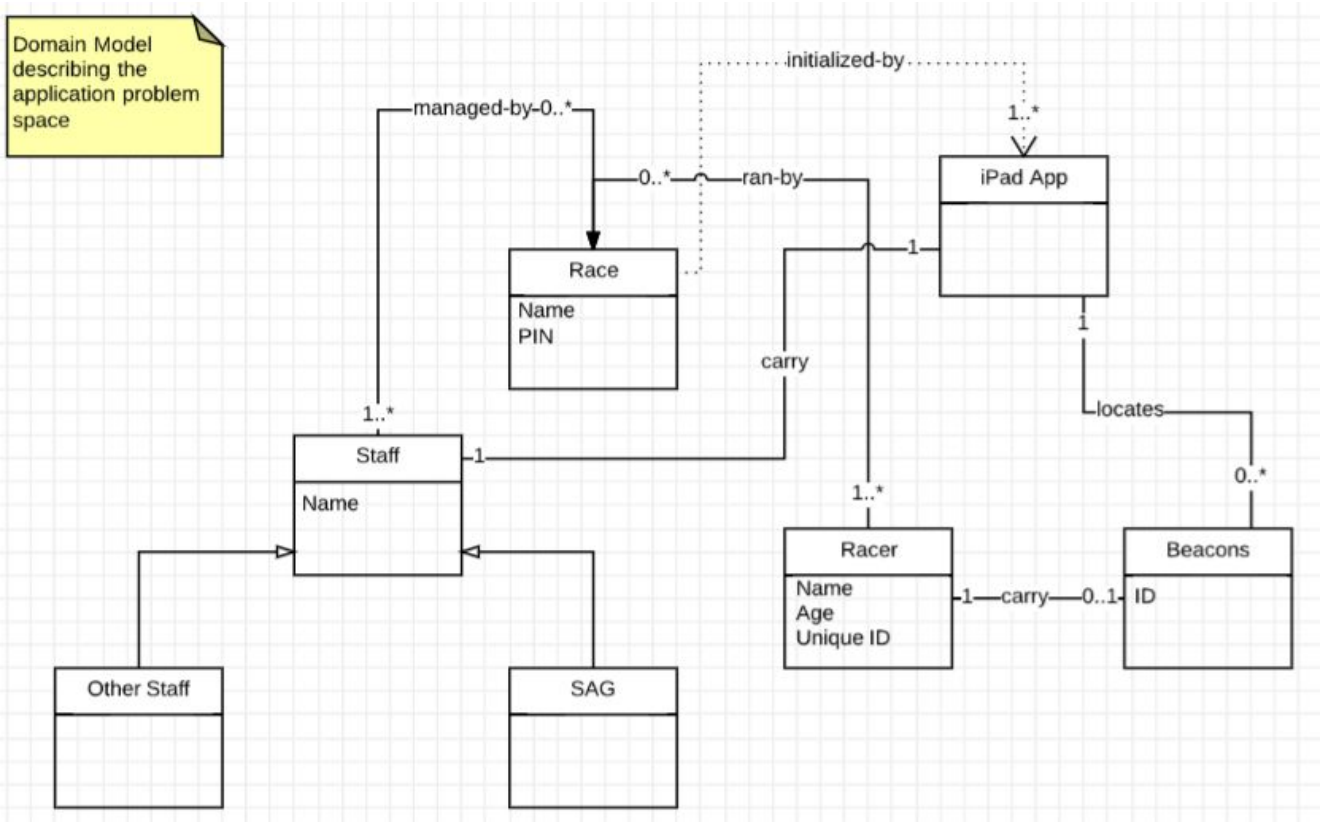
Advisor: Rick Weil
Sponsor: AJ Blythe

Team:
Danielle Neuberger
Randy Goodman
Anshul Kapoor
Tyler Schoen

Project Overview

- Race utility application
  - Locate and track status of competitors in race events via Bluetooth chips in range of iOS
- Variety of project components - web application, iOS app, bluetooth interaction
- Project had broad scope, specific use-cases
- Balance functionality with versatility
- No inherited codebase
- Increased emphasis on design
- Design decisions based on strengths

Project Overview - Domain Model



Project Overview - Goals

#	Goal	Priority
1	Ensure participants' safety during their activity	P1
1.1	Provide biker check in/check out	P1
1.2	Provide details of biker's status	P1
1.3	Estimate biker location/arrival information	P2
2	Provide beacon tracking in a standalone fashion, and capability to share that information over cellular networks as available	P1
3	Support concurrent events	P1
4	Allow staff to efficiently communicate with each other	P1
5	Enable the system to be configured for a variety of different event types	P2
6	Provide means for event promotion	P3

Project Overview - Non-Goals

#	Non-Goal	Reasoning
1	Virtual races	The main goal of the product is ensuring participant safety on a physical, centrally located racecourse.
2	Continuous tracking	Beacons are tracked only when proximal to a tracking device.
3	Fundraising Focus	This is not a fundraising event!
4	Route Creation	It's not meant to alter race routes (MapMyRide can do that), and it would be a separate project in its own.
5	Race Timing	Its meant to estimate the ETAs of racers, not time them.

Project Overview - Use Case 1

Name	Event Creation
Objective	To create events in preparation for a race
Pre-conditions	User has already registered an account on the website
Post-conditions	User has successfully created an event. Event is visible in the user's event list on the main page.
Primary Workflow	1. User signs into the web application 2. User clicks the '+' sign to be directed to the event creation page 3. User inputs event information to the event form and submits 4. User is redirected to the home page
Alternative Workflow	After step 4, the user may click the edit button on the row of the new event. At this point, they would be redirected to the edit event page and can make necessary changes.

Project Overview - Use Case 2

Name	Racer Check-in
Objective	To automatically check in a racer as they pass a checkpoint
Pre-conditions	1. The event was already created in with the web app prior to event start date 2. Event start date has not passed 3. Bluetooth chips have be assigned to registered participants 4. Staff member has logged into the iOS app on an iPad in a location on the race route
Post-conditions	1. Racer location updates on the iOS map 2. Racer status updates if necessary
Primary Workflow	1. Racer having a bluetooth passes the staff checkpoint 2. System automatically detects the bluetooth and updates racer location and status
Alternative Workflow	● If the rider takes a break at the checkpoint, the system will automatically detect that the rider has not left and will appropriately manage their status and location. ● Staff may also manually update rider status if necessary

Team Organization

- **Team Lead :** Danielle Neuberger
- **Testing Lead :** Randy Goodman
- **Documentation Lead :** Tyler Schoen
- **Development Lead :** Anshul Kapoor

Project Methodology

Spiral Model

- Hard deadlines and Agile don’t mix
- Iterative Development Process
- Continuous Delivery and Review
- Focus on Risk Analysis & Management

Derived:

- Internal Artifacts
  - Requirements Specification
  - Prototypes
  - Architecture Document
  - Project Plan
  - Test Plan
- Schedule

Technology Stack



Application

- iOS App
  - Requirement due to sponsor experience and perceived useable interface
- Sails.js
  - Fast framework on top of Node.js, similar to rails.
- MySQL
  - SQL-like database was a requirement
  - Chosen for popularity, good documentation, and ease of setup
- Mapbox
- Gimbal bluetooth beacons

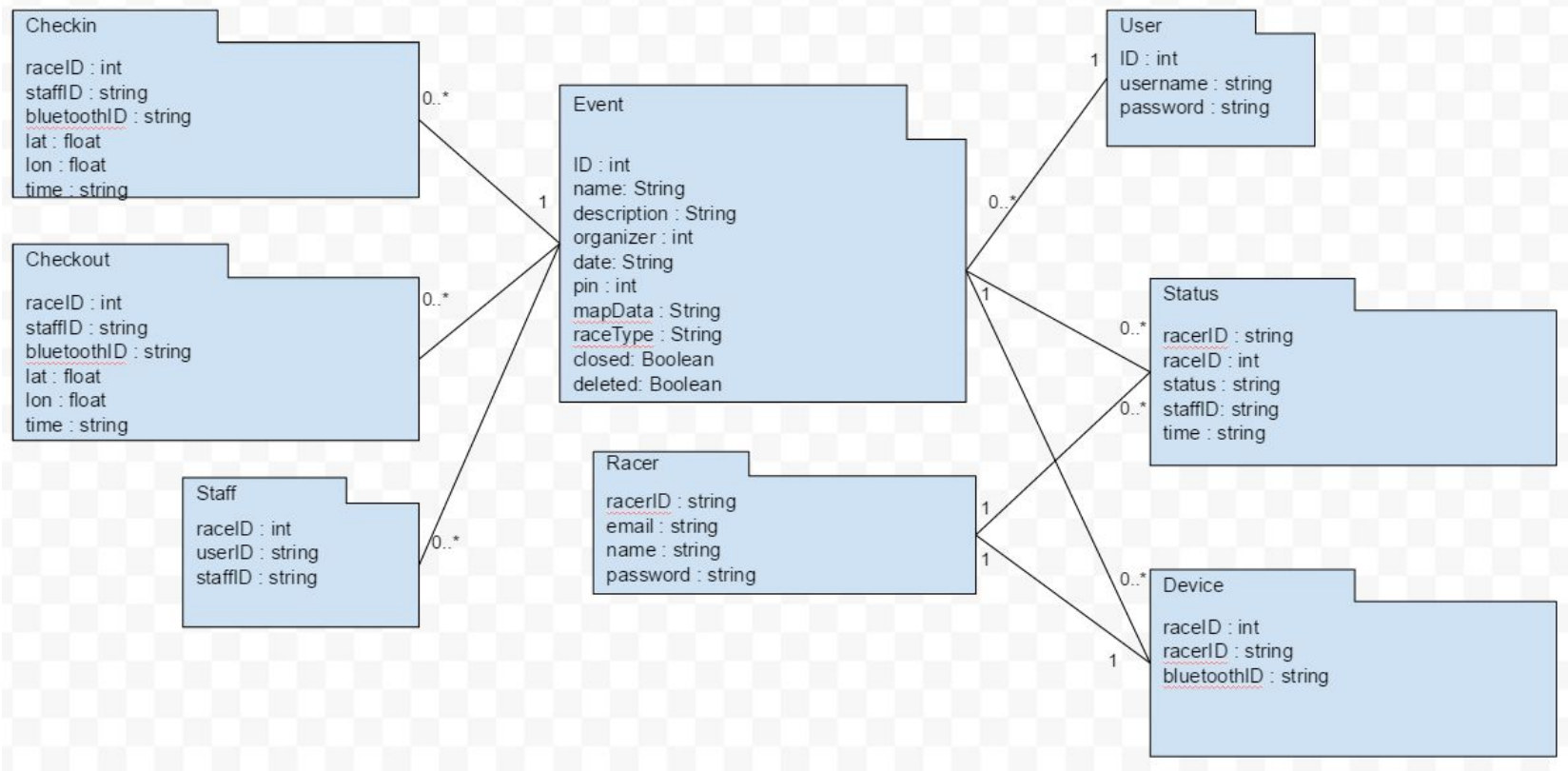
Development

- git/GitHub - version control
- Gantter - scheduling
- Slack - communication
- Waffle.io - task management
- Google Drive - document storage
- Jenkins - continuous integration

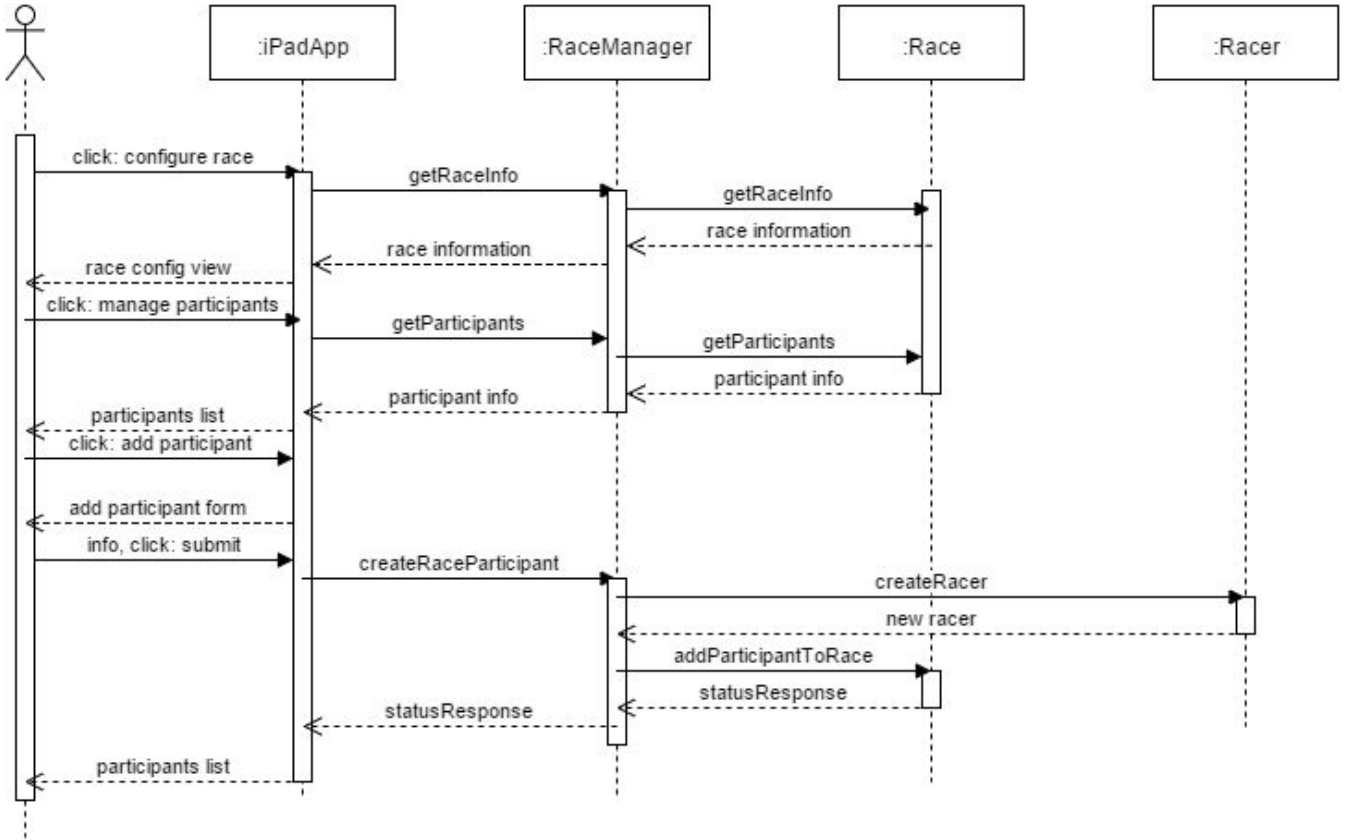
Testing

- mocha - test framework
- istanbul - test code coverage reporting
- superagent, supertest, request

Architecture - Data Architecture



Design - Associate Rider with Bluetooth



Current Status

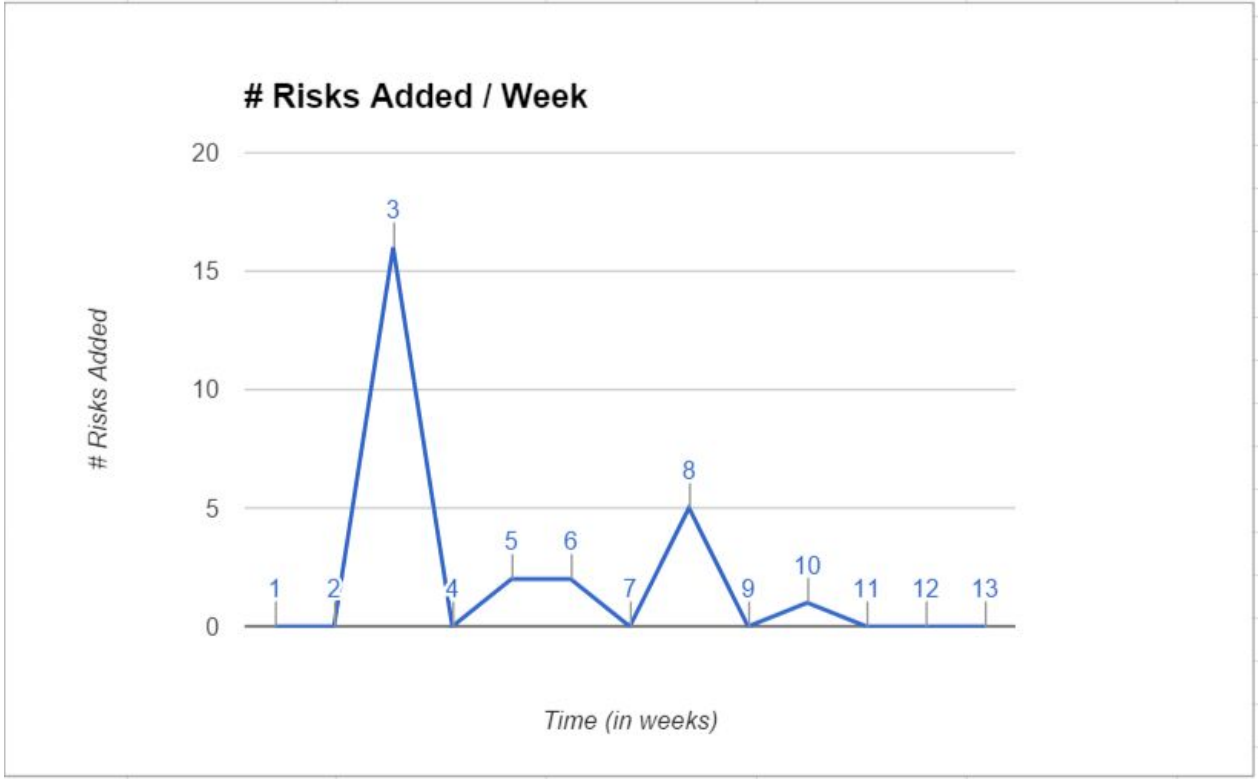
- Sails Application
  - Registration, Log in/out
  - Event CRUD
- Jenkins running tests and building
- Sails Application and MySQL server live on a SE Department provided VM
- iOS App
  - Beacon Check-in
  - Add Racer
  - Manual Status Change
  - View Race map
- ~1/3 through P1 Requirements dev

Metrics

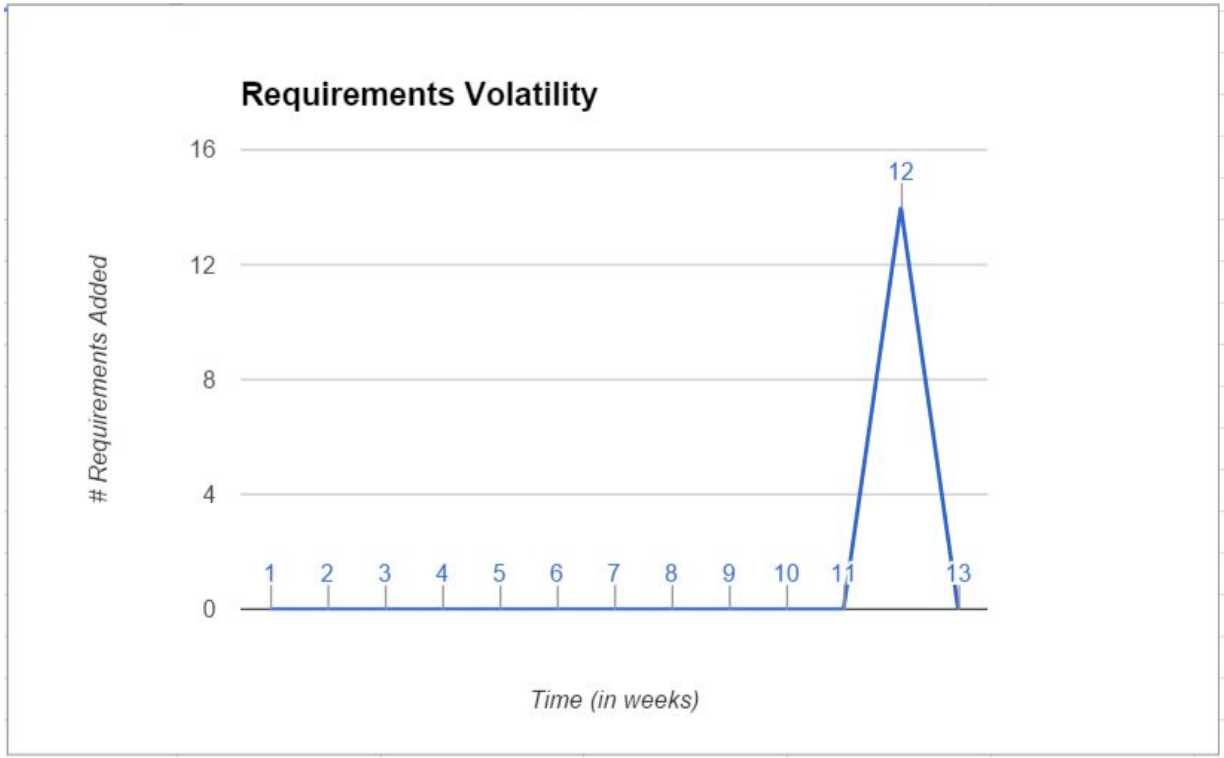
- # bugs/SLOC
- % program statements called during test suite execution (statement coverage)
- # commits per week
- # risks added per week
- % relevant risks
- [unofficially] requirements volatility

Metrics - # Commits / Week

Metrics - # Risks Added / Week



Metrics - Requirements Volatility



Risks

Risk ID	Risk	Description of Risk	Probability (0 - 1)	Impact (low, med, high)	Exposure	Classification
3	Inadequate bluetooth experience	None of the team members have an expertise in BLE beacon development, which may take time.	1	high	10	quality
2	No experience with iOS or Objective-C/Swift for Mobile application	Team members have very limited experience with iOS development	0.8	high	8	quality
4	Difficulty connecting components	Connecting components representind various features may be difficult	0.65	high	6.5	quality
5	Manufacturing defects in bluetooth beacon	There may be certain issues with the hardware that the team is currently unaware of.	0.6	high	6	quality
6	Choosing inappropriate/inadequate tech stack	The team doesn't have strong experience in mobile development, due to which we may choose an inefficient tech stack.	0.5	high	5	quality

Demo

Future Development

- More features
  - Racer visibility on map
  - Staff chat
  - Racer ETA & missing notifications
  - Racer/public web-app interface
  - Superuser web-app interface
- Enhanced testing
  - Unit, Integration, Usability

Reflection

What went well:

- Collaboration
- Communication
- Time dedicated towards project
- Prototyping

What didn’t go well:

- Initial long iterations
- Initial poor workload estimation per iteration
- Mapbox SDK bugs

Questions