Trip Planner

Mohammed Reda Ait Hammouda

Trip Planner Overview

- What is it?
  - Web application that will aid users in gathering information about destinations including country/state/location information, airport information, and attraction information
  - Interactive map where users can hover/click into a country/state/location to get high-level and more detailed information
  - My Trips where users can build and save trip plans
    - Attractions
    - Cities
    - Travel Plans

Why I chose this?

- Anyone who knows me knows that I love to travel as much and often as time and my bank account allow
- I have little experience with front-end development/technologies. I see this as an opportunity to try my hand at learning these skills and attempting to build a cool product at the same time that aligns with my favorite hobby
- I am looking for other travel enthusiasts that either have experience with front-end web applications or a strong desire to dive in and learn alongside me!

Possible Technologies

- I am envisioning this as a web application hosted in the cloud (aws)
- Will utilize a couple travel/map apis and libraries to get real-time data to build out the interactive map and location details page
  - Mapeal
  - Triposo

- HTML
- CSS (boostrap)
- JS (Vue or React?)
- MongoDB
- Travis CI
- Unit.js for unit testing
- Git/Github
- Budibase
- Aws

Nice-to-have Functionality

- If time allows, I would like to have automated functional tests utilizing:
  - Selenium
  - Gherkin

Sprint 1: Review and Retrospective

Mohammed Reda Ait Hammouda

Sprint 1

Focus and Accomplishments:

- Presented application and the vision he had in mind with the team
- Created and presented mock-ups of the web application
- Completed spikes on AWS as a cloud solution to host our application
- Completed spikes on apis that can be used for interactive mapping

Using the information learned from the spikes and research, we were able to create a minimum viable product design and the user stories to accomplish that vision

Tools and Technologies

- Communication: Slack
- Scrum Board: Trello (https://trello.com/b/AIHUWG4I/se491-se-capstone)
- Source Control: Github (https://github.com/tylormondragon/SE491_Capstone)
- Document Management: Google Drive
- Languages: HTML, CSS (Bootstrap), Node.js, JavaScript
- Hosting/Monitoring: AWS Elastic Beanstalk
- Estimation: Planning Poker
- CI: TravisCI?
- Maven: Build/Packaging?
- Unit Testing: Jest/AVA framework?
- Functional Regression: Selenium (NTH)

Rough Project Timeline

Sprint 1:

- Determine technology stack
- Determine project management/communication tooling
- Sprint planning and user story creation
- Story estimation and velocity

Sprint 2:

- Main page HTML/CSS implementation
- Detail page HTML/CSS implementation
- Main page JS framework implementation
- Interactive Map implementation

Sprint 3:

- Detail page JS framework implementation
- Location Search (Sygic Travel API)
- Map Hover Functionality
- Interactive Map implementation - Other Countries

Sprint 4:

- Bug fixes
- Additional Testing
- Domain Creation
- AWS Hosting, Running, Monitoring
- Carry over due to expected technology learning curve

Writing Epics, Stories and Tasks

Grouped stories into 6 Epics and have planned stories for development

1. Find Location
2. Responsive UI Design
3. Testing
4. Trello/PM Management
5. AWS Setup & Management
6. Skill Building Allocation

User stories and tasks created accordingly (see Trello board)

Estimation

- 1 - Less than a day
- 2 - 2 days
- 3 - 4 days
- 5 - 1 week
- 8 - 1 week to 2 weeks

Utilized online poker planner to decide on story points for User Stories and Tasks

Assuming we can complete 35 story points a sprint / ~7 per person for each of sprint 2-4

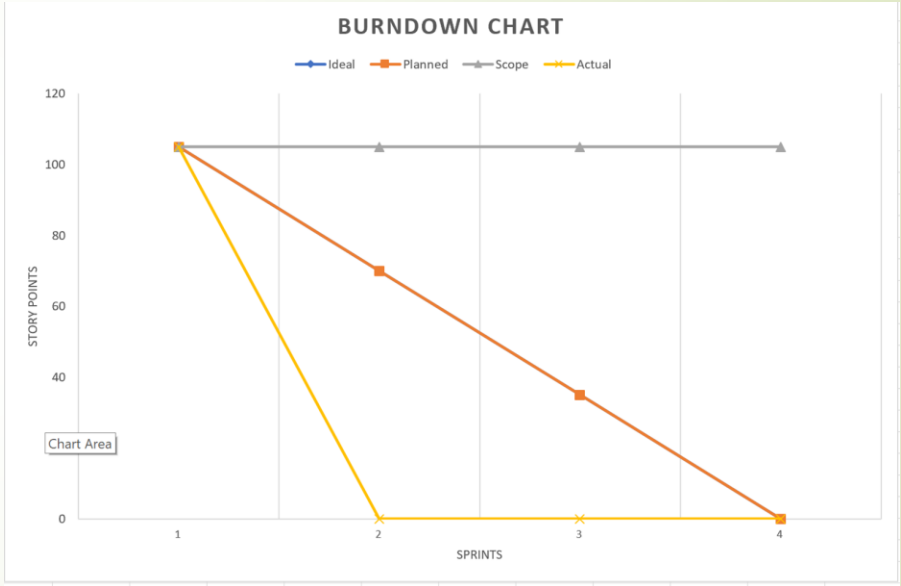
Definition of Done

- User story has a all commits pushed to github/pull request created and reviewed by another team member
- Continuous Integration runs/passes before a commit can be merged
- Commit is merged into master in source control
- Unit tests, where applicable, are implemented and passing 100%

Project Plan

Story Points...	S2	S3	S4
At start of iteration	105		
Planned done during iteration	35	35	35
Actual done during iteration			
From changed estimates			
From new stories			
At end of iteration	70		

Sprint Burndown



Retrospective

Start:

- All team members communicating through slack regularly
- Give regular updates to the slack group so we can track/monitor progress
- Have 2-3 “stand-up” meetings a week to keep project on track
- Adding more detail to the user stories/tasks
- Working on tasks and moving tasks to appropriate lanes on our scrum board

Continue:

- Listening to every member and adjusting things accordingly/being flexible
- Individual learning of the technologies/languages as they are all new to us
- Being fun to work with and creating a welcome and enjoyable team culture

Sprint 2: Review and Retrospective

Mohammed Reda Ait Hammouda

Sprint 2

Focus and Accomplishments:

- We successfully created our node.js base setup
- Created the landing page UI and some of the details page UI
- Configured application on AWS using beanstalk and pipelines
- Got API keys to a couple different travel APIs and spent time playing with those to get familiar how to use APIs and JavaScript

[UPDATE]: we did get these working successfully right before class started!
What we did not quite accomplish:

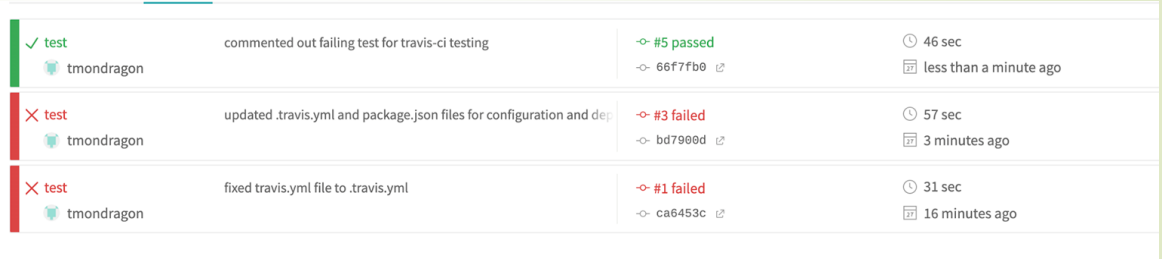
- Attempting to connect project with travis CI, but not having success getting it to work with project
- Attempted to set up demo unit test, but not having success getting it to work - need additional research in this coming sprint

Technologies

New this sprint:

- Travis CI
- Jest

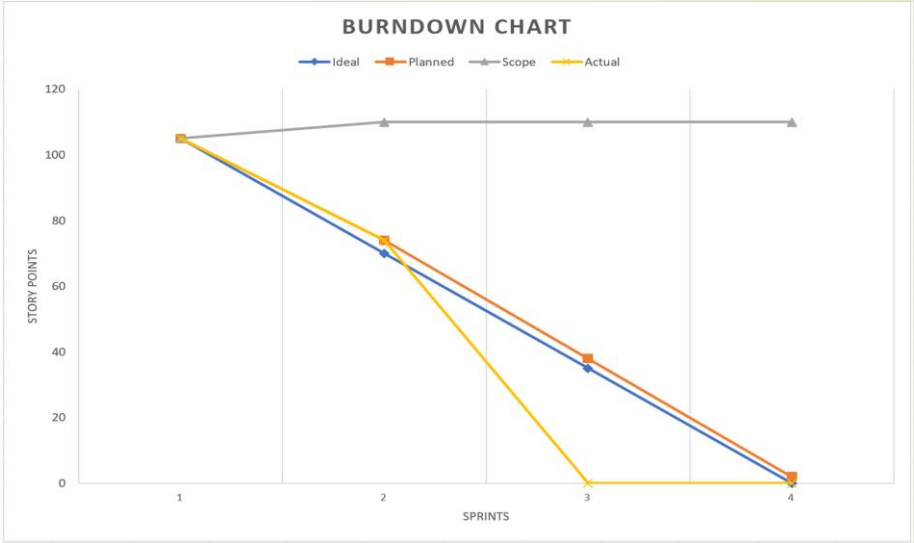
```
223
224  PASS  tests/trip.test.js
225   ✓ Hello World (1ms)
226
227 Test Suites: 1 passed, 1 total
228 Tests:       1 passed, 1 total
229 Snapshots:   0 total
230 Time:        0.905s
231 Ran all test suites.
232 The command "npm test" exited with 0.
233 store build cache
248
249
250 Done. Your build exited with 0.
```



Project Plan

Story Points...	S2	S3	S4
At start of iteration	105	74	38
Planned done during iteration	35	36	36
Actual done during iteration	36		
From changed estimates	5		
From new stories	0		
At end of iteration	74	38	2

Sprint Burndown



Sprint Demo:

http://se491capstone-env-1.eba-w8xbpmgp.us-west-2.elasticbeanstalk.com/

Retrospective

Stop:

- Pushing directly to master (github was not configured correctly this sprint)
- Writing all of the code in a single .js/.html/.css file
- Scheduling meetings last minute
- Not pushing code up as you work - hinders other users depending on that code

Start:

- Having group/pair programming sessions
- Creating branches and doing pull requests for any commit to get into master
- Having code reviews by another team member
- Regularly using developer tools console/network to watch for any issues with JS
- Schedule all video meetings during sprint planning meeting

Continue:

- Supporting team members new to using git for push/pull/merge issues
- Being a fun and supportive team to work with

Sprint 3: Review and Retrospective

Mohammed Reda Ait Hammouda

Sprint 3

Focus and Accomplishments

- Design and implement country-level details page
- Design and implement city-level details page
- Add interactive clickable world-map to main page
- Work on api mocks (still in progress)

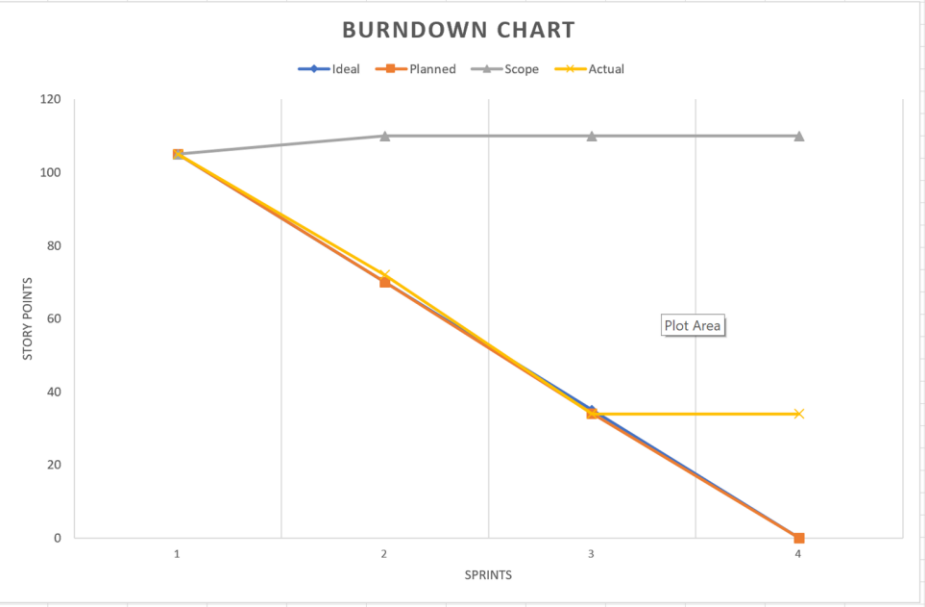
Other:

- We ran into an issue with cross-site cookies (we think related to Chrome) and was not able to get the issue fixed so data populates all dynamically

Project Plan

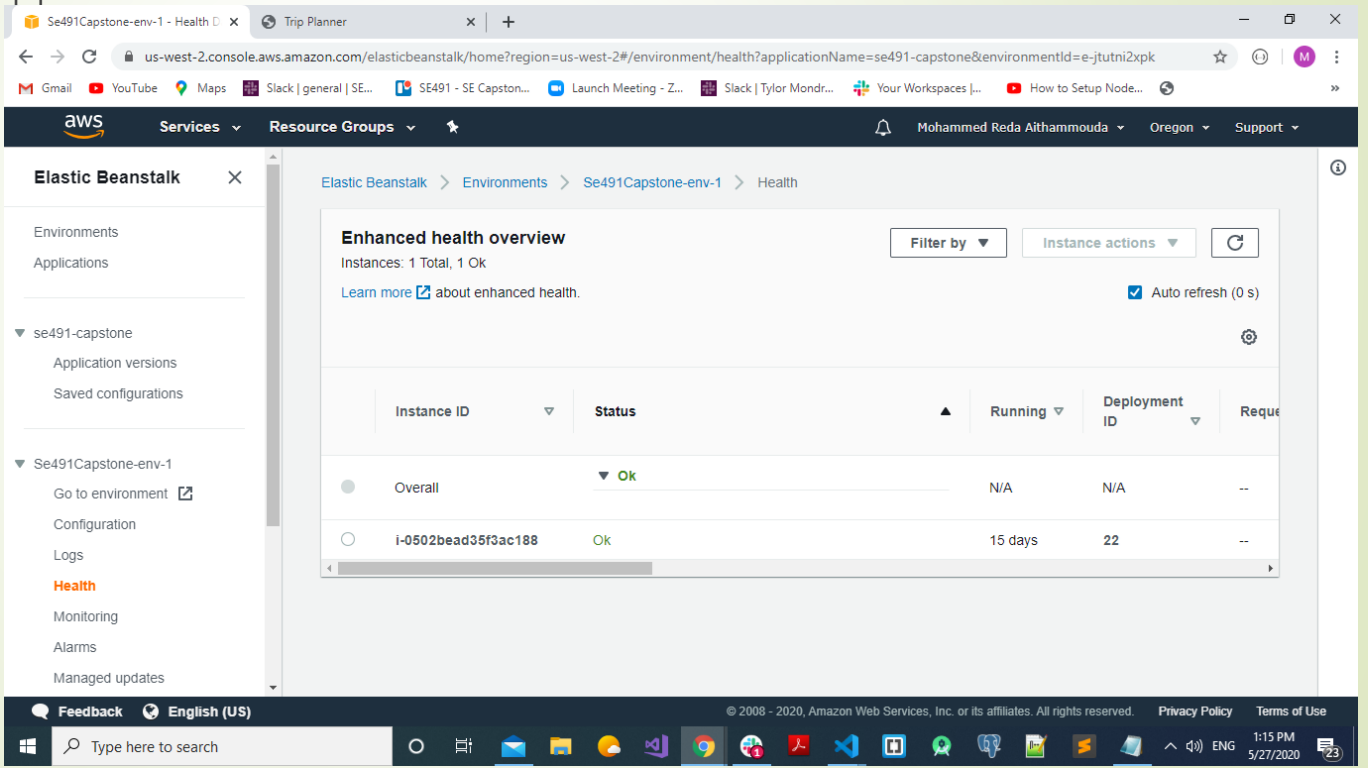
Story Points...	S2	S3	S4
At start of iteration	105	72	34
Planned done during iteration	35	36	34
Actual done during iteration	38	38	
From changed estimates	5	-5	
From new stories	0	5	
At end of iteration	72	34	0

Sprint Burndown



Project Health & Monitoring

> We utilized AWS functionality for continuous deployment and for monitoring the health of our application



Retrospective

Start:

- More communication, especially around issues
- Cross-Site issue three people had independently but didn't raise to the group
- Updating tickets on trello board as you make progress

Stop:

- Duplicating work that is already being done
- Not merging code more frequently

Continue:

- Pair/group programming exercises
- Being a fun group to work with

Sprint Demo:

http://se491capstone-env-1.eba-w8xbpmgp.us-west-2.elasticbeanstalk.com/

Sprint 4: Review & Retrospective

Mohammed Reda Ait Hammouda

Sprint 4

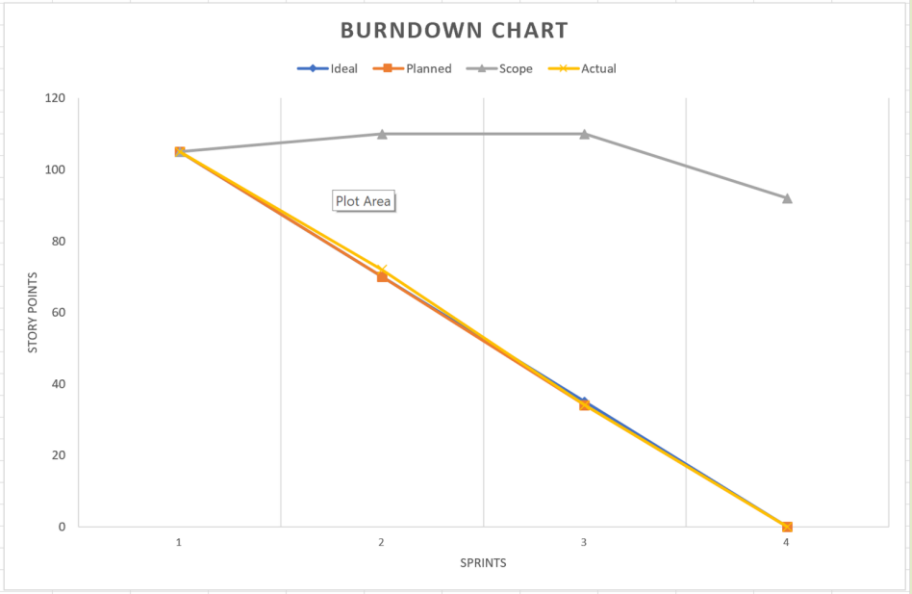
Focus and Accomplishments:

- Fix bug so dynamic functionality works
- Add mapbox map to attractions page pop-up
- Add attraction details to pop-up
- Search by countries on main page
- Explore countries by continent
- Tie up current vision to end quarter

Project Plan

Story Points...	S2	S3	S4
At start of iteration	105	72	34
Planned done during iteration	35	36	34
Actual done during iteration	38	38	16
From changed estimates	5	-5	0
From new stories	0	5	-18
At end of iteration	72	34	0

Sprint Burndown



Disaster Recovery

- Identify and establish database backup
  AWS backup service
- Ensure API layer is well written so code is API agnostic in the event one stops working or a
  change to another provider is needed
- Identify AWS region failover in the event the active host server goes down the site can remain
  online with little to no downtime
- Long-term work on building out our own API and database with travel information to
  eliminate dependency on third-party providers

Sprint Demo

http://se491capstonebeanstalk-env.eba-k4xmw3du.us-west-2.elasticbeanstalk.com/

Retrospective

Start:

- Begin testing immediately with development
- Possibly consider test-driven development approach (TDD)
- Get familiar with applicable test framework at beginning of project
- Commit often!
- Push up small commits and do not try and push up all changes at once
- Have more frequent in-person stand-up meetings (~15 minutes)
- Have CI configured at beginning of project to encourage test writing early on

Stop:

- Do not duplicate work
- Make sure tasks/user stories are clearly defined
- Working on features/functionalities not in the sprint backlog
- If time allows, address this, but try not to start working on functionality withouthaving discussed with group

Continue:

- Pair/group programming
- Effective to learn new technologies and add a second pair of eyes for review

Sprint Demo

http://se491capstonebeanstalk-env.eba-k4xmw3du.us-west-2.elasticbeanstalk.com/