SEEK NYC

ALPHA TEAM

Close-out Report

June Pham

Dewitt Johnson

Honda Huang

Mark Won

Justin Lam

Executive Summary

Our goal with Seek is to streamline the process for seekers to better interact with the local disaster shelters in New York City. We want to create a reliable system for the residents of NYC in the event of a natural disaster, such as hurricanes and blizzards. We hope to provide a simple and intuitive interface for users from all backgrounds to search and register for shelters. We will ensure that our database is regularly updated and maintained to uphold the stability required for an emergency service.

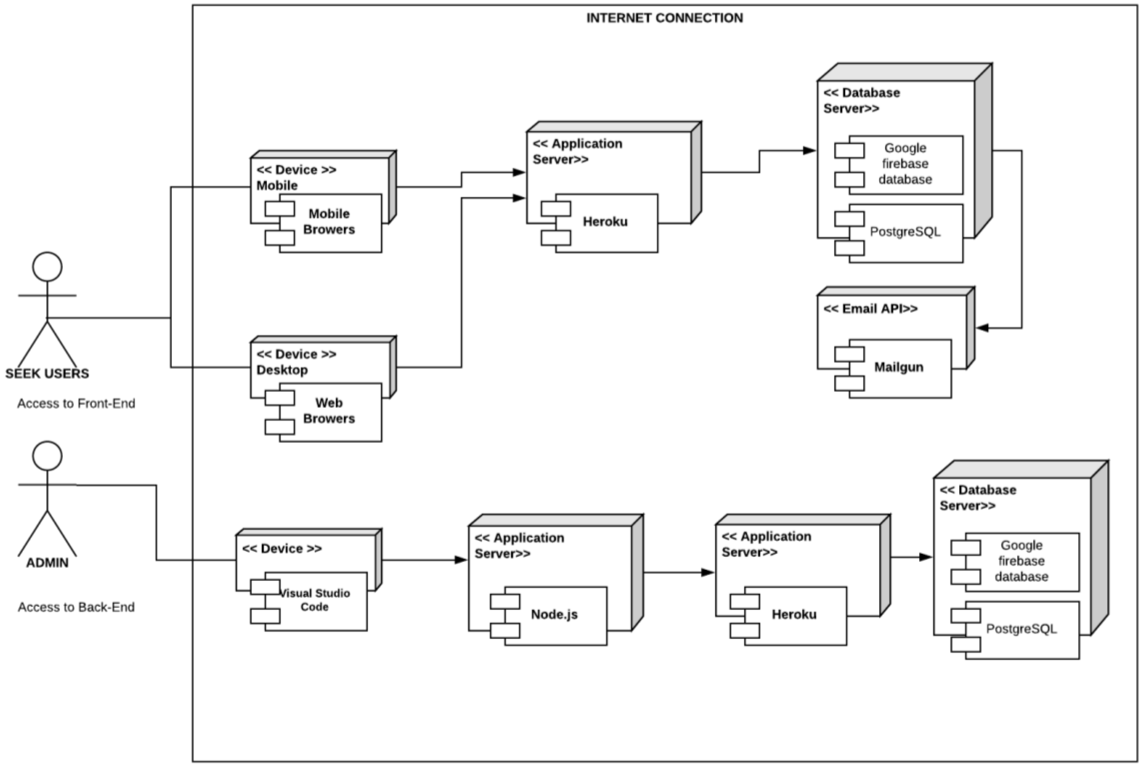
We at SEEK have successfully entered the closing stage of our project. Our website is fully functional and meets all the user requirements that we have set our at the beginning. We have completed both front-end and back-end development of the website. The site functions properly in displaying the various NYC disaster shelters and the form submission page is now integrated with the database. We have also completed all quality tests for both the front-end and back-end. In this closing stage, we will be going over our deployment programs and lessons learned.

Deployment Plan

1. **Hardware/Software:**

	Resources	**Cost**
Hardware	Server & Web Hosting	$50 per month
Software	Flask, ReactJS, NodeJS, PostgreSQL	Free
Storage	PostgreSQL	$200 per month
Network	Internet	Provided with web hosting
Product environment	Heroku hosting	Free
End User(s) Environment	Web Application	Free or pay for development
Development Environment	Software Developers	$11,750

2. Deployment Diagram



3. Training Plan

In order to maintain functionality of the site, we must train the shelter workers to be able to update any important changes relevant to the shelters. We will have our DBA visit the various shelters and give them access to the database along with teaching them how to make the appropriate changes. We must also train the support staff to be able to handle any inquiries regarding the site and its interface. Lastly, we will need to train a database worker to ensure all shelter information are up to date and to make any necessary changes to reflect the shelters in case anything were to happen to the database. Our training resource would include the budget and the current staff to complete our training regiment.

4. Support Plan

We plan to use a content delivery network to improve the accessibility of our site. We will also have multiple host servers as backup to ensure the site remains running at all times. This may include having servers hosted in others states in case the servers in NYC go offline. Our system support will mainly revolve around our IT Managers maintaining the site.

5. Enhancements

In the future, we would like to expand upon on site to include not only disaster shelters but also homeless shelters. We are also planning on expanding to include all shelters located throughout New York State. We may also consider a mobile app that is preinstalled on all NYC phones by default as a safety measure. We also hope to expand on the awareness of our product so that the residents of NYC will know where to turn to, in the event of a disaster. By having our site listed on other government agency sites and local community sites(such as schools and churches), we will be able to effectively raise awareness.

Closeout Report

1. Updated Requirements Traceability Matrix

Req. ID	Requirement Description	Functionality	Use Case ID	Use Case	Priority	Complexity	Requirements Met
R1	Database should update capacity of shelter in near real time	Functional	UC1	Update	High	Moderate	Yes, the database contains and updates the current number of occupants in each shelter when information is pushed or pulled
R2	System should store and return location of shelters	Functional	UC2	Stored Location	High	Simple	Yes, the location of the shelters are stored and displayed on the site
R3	User should be able to get directions to shelter	Functional	UC3	Directions	High	Moderate	Yes, the user is able to click on the pin of a specific shelter to view the directions through google maps
R4	User can reserve a spot in a shelter of their choice	Functional	UC4	Reserve	High	Moderate	Yes, the form submission page is running and the integration with the database has been completed
R5	User should be able to look for a shelter	Functional	UC5	Seek Shelter	High	Moderate	Yes, users are able to view the various NYC shelters on the google maps interface using the search bar or the

							interface itself
R6	System should link to Google Maps to display shelter locations	Non Functional	UC6	Maps	Medium	Moderate	Yes ,the google maps api is linked to the homepage of the site
R7	System should be able to handle large amount of users	Nonfunctional	UC8	Load Bearing	Medium	Complex	Yes, the system can handle a large amount of users
R8	Database info should be current (1 month)	Nonfunctional	UC9	Up-to-date	High	Simple	Yes, all shelters in NYC are included in the Database, all current users are in the Database
R9	The database must be able to recover or restore to its data of no more than 1 hour before system failure occurred	Nonfunctional	UC10	Recover	Low	Complex	TBD, not yet tested
R10	Website should be mobile-friendly	Nonfunctional	UC11	Mobile	Medium	Simple	Yes, site is functional on mobile and has different CSS for different media screens

2. Updated Test Cases

Test ID	Test Name	Test Description	Pass/Fail/ Not Tested	Requirement on traceability matrix
TC1	Reserve	Users can reserve for a spot and are entered into the database.	Pass	R4
TC2	Accurate Directions	Appropriate directions are displayed	Pass	R3
TC3	Responsive	Site should scale to various screen sizes	Pass	R10
TC4	Search Shelter	User should be able to search for nearby shelters	Pass	R5
TC5	Received Notification	System should successfully send the User notifications	Pass	R4
TC6	System Updates Database	Update Database Use Case	Pass	R1

TC7	System Determines user's location	Determine User Location Use Case	Pass	R3
TC8	System stores location	All shelter locations show on map	Pass	R2
TC9	User Checks into the Shelter	Check In Use Case	Pass	R4

3. Product/Project tables

Project/Product Measurements Product Acceptance Criteria/ Project Success Criteria/ Project Objectives	**Description**	**Met/ Not Met**	**Comment (indicate how Met/ why Not Met)**
Google Map API	The incorporation of Google Map with all the disaster/hurricane shelter spots on the map. The map will automatically zoom it into the person's location once the user allows for location usage. The user will be able to get directions to the shelter.	Met	We integrated the Google Maps API onto our homepage and listed the various shelters along with their locations as pins on the interface. We also included a link to maps.google.com which showed the directions to the individual shelters. We used javascript to obtain the user's location.
Search Function	User is able to search for a shelter	Met*	We were able to meet this requirement however due the time constraint and servers crashing, the search bar was removed in the process as we were preoccupied with integrating the back-end with the front-end.
Reserve Form/ Check-in Form	User will be able to reserve for a shelter spot through the website. User will be able to check-in to a shelter to eliminate traditional paperwork	Met	We created the check-in page by using HTML, CSS, and Javascript. We were able to get the forms to work when the database was finally connected.
Email Notification	User is able to receive an email notification once the check in or reserve a spot	Met	We store the user's email onto to our database where the system proceeds to send a predesigned email to the user the moment the form is submitted using
Check Shelter Capacity	User will be able to select a shelter and see the capacity and shelter spots available	Met	We included the capability to our check-in page by adding HTML which listed the shelters and its respective information.
Responsive Website	The website can be viewed on multiple devices	Met	We included different CSS for different screen types and sizes so that the site is responsive and functional across all devices.

Contact & About Page	Contact and About page lists out the functions of the website and contact information	Met	We created the About page by using HTML and CSS. We listed the various information about our site along with contact information at the bottom of the page.
Database Integration	Database with shelters capacity integration	Met	Database is created with PostgreSQL and is hosted on Heroku. Integrated with webapp using Python.

4. Scope Changes

There wasn't any major change from the original plan we had set out for ourselves with the SEEK NYC project. But we did extend our scope because the prior scope wasn't enough for us to complete the project within the allotted time, according to our Project Management Plan. We attempted to add the admin page midway to allow the shelters to be able to log-in and change the capacity and resources of the shelter but it was beyond the scope that we had planned and decided to not include it in the closing project.

Lessons Learned

1. Describe the project area(s) that the team did well

The project areas that we did well in are cost management, stakeholder management and communications management, risk management and quality management. The project was completed within the planned budget. We also did well in the delivery of project milestones to stakeholders. We initially had issues with team communication, but that was solved quickly and effectively so that project requirements weren't being impeded. We did a good job in anticipating the known risks of the project and did our best to alleviate the issues that occured. The project expectations were delivered within major milestone deadlines and were mostly consistent to what was projected to be delivered.

2. Describe the project area(s) that the team did not do as well

The project areas that we can improve on are human resource management, time management and scope management. We could have improved human resource management in better gauging and understanding the strengths of each individual so the team can help work on all tasks when needed. Time and scope management ties closely together in how we had to adjust the scope of our project to meet the deadlines we faced. We had planned for potential known risks (database integration) and allocated extra time for that in comparison to other tasks, but underestimated the actual difficulty of the task.

3. Describe any take away and/or feedback of working on the project

Team communication is incredibly important for the completion of a successful project. No matter how well a project is planned it is difficult to anticipate all the risks. Deliverable presentations to stakeholders are good for getting insightful feedback.

4. What would be done different if placed on another similar project

Project scope and time will be better adjusted for because we have a better understanding of which tasks are more problematic than others. Try and have more in-person meetings to work on the group project together. We were able to get a good amount of work done when working together in-person. Problems and questions were answered much quicker and it was easier to work on a task together.

Project Selection Methods

We have picked four methods to analyze and select our project for this semester.

1. **Focusing on broad organizational needs**
   a. **Needs:**
      i. To streamline communication among shelters in New York City.
      ii. To provide better access in terms of database to existing shelters.
      iii. To minimize congestion of shelters during crisis.
      iv. To improve safety & efficiency among shelters.
   b. **Will:**
      i. Natural disasters and hurricanes are happening more frequently than ever due to global warming thus we need to provide a solution for shelters to effectively help people and minimize capacity bottleneck in the process. .
   c. **Funding:**
      i. Funds will be given by the nonprofit organizations or by investors that are interested in our project initiative.

2. **Categorizing IT projects**
   a. **Opportunities:** Focusing on shelters help our team to learn more about nonprofit organization structure and the corporate social responsibility behind it.
   b. **Problems:** During winter and hurricane season (June - July) there are an influx of people into the shelters and we want to relieve that problem with our system
   c. **Directives:** Gather more requirements from the government and nonprofit organizations that we work with in consideration of legality and confidentiality of users
   d. **Deadline:** 12 weeks
   e. **Category:** High Priority Project

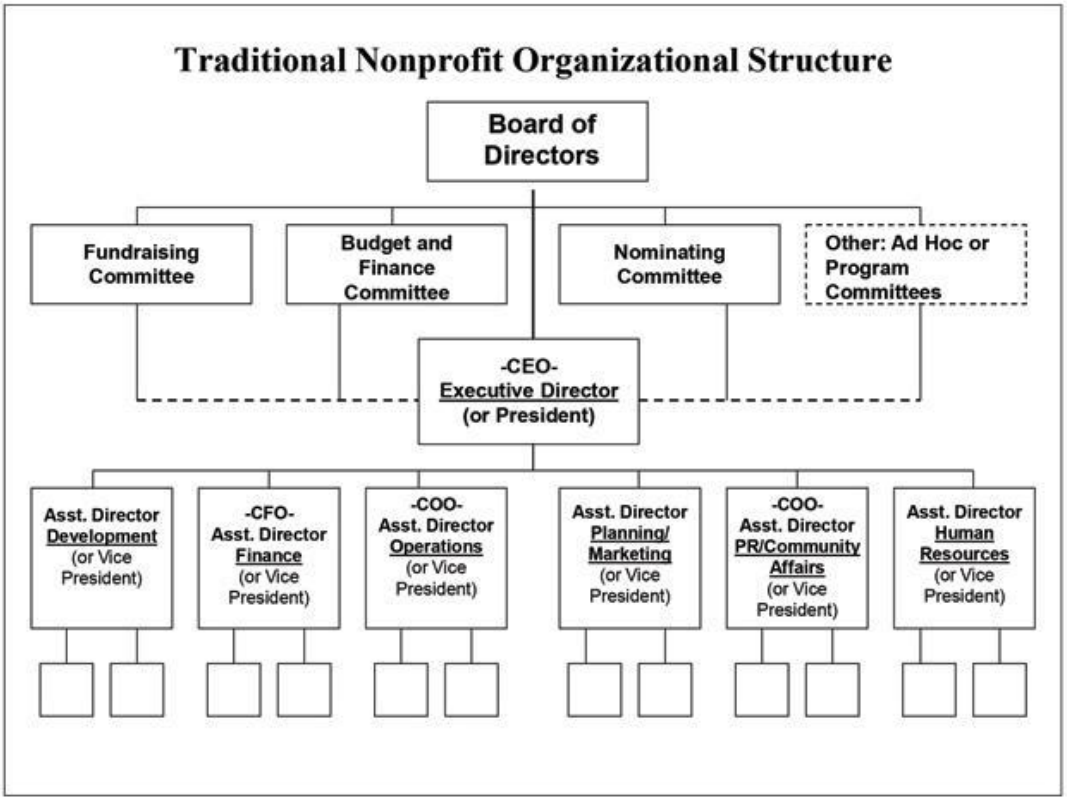
3. **Using a weighted scoring model (We compared this idea to the inventory video game idea that we have also considered)**
   a. Supports key business objectives or strategies in corporate social responsibility for our team
   b. Has a ready built infrastructure (computers for employees to use & internet access)
   c. Can be implemented in approximately 12 weeks
   d. Moderately low risk in meeting scope, time, and cost goals
   e. Strong team who believes in the idea and possess the necessary skillset

4. **Implementing a balanced scorecard**
   a. **Financial:** Lower cost → More profit, effective use of resources such as our team skill set and budget proposed.
   b. **Internal Business processes:** Makes process more efficient and streamlined.

c. **Learning and Growth** Improve team's morale, learning more about shelters and nonprofit organizations

d. **Customer:** Satisfied customers and stakeholders.

Organizational Framework


Traditional Nonprofit Organizational Structure
Board of Directors
Fundraising Committee
Budget and Finance Committee
Nominating Committee
Other: Ad Hoc or Program Committees
-CEO- Executive Director (or President)
Asst. Director Development (or Vice President)
-CFO- Asst. Director Finance (or Vice President)
-COO- Asst. Director Operations (or Vice President)
Asst. Director Planning/ Marketing (or Vice President)
-COO- Asst. Director PR/Community Affairs (or Vice President)
Asst. Director Human Resources (or Vice President)

Source:(http://www.hurwitassociates.com/nonprofit-organizational-charts/traditional-nonprofit-organizational-structure)

Our team will focus on developing a website backed by a robust database containing emergency shelter locations in the event of a disaster. Our project will be sponsored by the New York State government as the website will be available for anyone residing within NYC. This is a matrix organization framework where employees often report to one or more bosses. Shelters that we are going to work with are mostly non-profit organizations receiving grants from the government. An employee within a non-profit could possess multiple roles and take on different responsibilities. But a framework of a non-profit is to not make any money by the end of the yearly quarter, they must allocate their money to either sponsoring our project or to other initiatives that they have taken interests in.

Project Charter

Project Title: Seek

Date of Authorization: February 7th, 2018

Project Start Date: February 9th, 2018

Projected Finish Date: May 4th, 2018

Time: 12 weeks

Budget: $25,000

Vision:
We want to create a website to better connect people in need and shelters at a time of disaster. Our users will be able to easily find a shelter with open capacity and be able to get the directions to the shelter in the event of a disaster.

Helping people in areas prone to disasters, limit the loss of lives, improving shelters nearby eliminating evacuation bottleneck. We want to create a website with a database that allows people to search for shelters nearby their location, get the directions to the shelters and register for a spot.

Objectives:

1. Creating fully functioning website that is integrated with the database of available shelters
2. Shelter seekers should be able to:
   a) Find the nearest shelter
   b) Get the directions to the shelter
   c) Reserve their spot at the shelter
   d) Mark/Flag themselves as safe
3. Shelter employees should be able to:
   a) Manage capacity
   b) Manage multiple locations (assuming they work there)
   c) View user information for those registered
   d) Flag any users that are in need of medical attention
   e) Request resources when low (food, water)

Mission or Approach

a) First team meetup to discuss roles and project goals and ideas
b) Initial project planning and task delegation

c) Analysis and finalization of the project plan
d) Initiate design on project
e) Create UML models and diagrams
f) Begin development of the website
g) Continue development of website
h) Integrate database into the website
i) Wrap up coding and development
j) Testing of project
k) Implement software
l) Finalize project and review database into website

Success Criteria

a) Meeting project objectives and deliverables
b) Internal and External stakeholder satisfaction
c) Meet proposed project budget, time and scope
d) Streamline communication between user and shelters
e) Less influx of people to shelters during the busy season
f) Easy to use and seamless user experience

Critical Assumptions and Constraints

a) Time frame might not be enough to meet all of our objectives
b) We only have 5 members
c) Project scope may need to be adjusted to the time constraint

Stakeholder Register

Internal Stakeholder	External stakeholder
Project Team (us)	External Users
Top Management	Competitors
Project Investors	Communities
Project Sponsors	Rudolph Brown
Employees at the shelter	
Internal Users	
Government	

Technical Approach

Our team plans to implement the use of agile scrum methodology to develop our project. We plan on starting with the initiation of the project, having the project charter completed and then approved by Chief Director of Operations, Professor Brown. Our next step is planning out our project plan by gathering the necessary requirements. We then plan on moving forward to the execution of the project where we will design the front and back end of our website. Before moving on to the final step, we plan on testing the website to eliminate any bugs along with meeting the requirements we set up in the planning phase. With everything complete and working as intended, we will be able to formally close the project

Technology/tools:

- Heroku (hosting application)
- Amazon Web Services
- cPanel
- MySQL

We plan on utilizing the above technology to design both the front and back end of our website. Our front end will be designed using a combination of HTML, CSS, and Javascript along with the help of Heroku. Our back end will mostly be developed using SQL in the built in MySQL application on the website's cPanel.

Techniques:

We will use techniques such as expert judgment and meetings to move our project forward. Using the available information online, along with our knowledge areas in process group management, we will use expert judgment when a problem without a clear solution arises. Hosting meetings will allow the project team to collaborate and aid each other throughout the project's life cycle along with staying on track of deliverables. We will also make use of best practices when applicable. Using SWOT Analysis and SMART requirements will ensure that our project successfully meets the goals we set.

Project Management Plan (WBS)

Budget = $35,000
Time: 2/10/18→ 5/10/18
Initiation ($2,000)

- Define project main objectives
- Identify internal and external stakeholders
- Create Project Charter
- Project Charter review
- Project approval

Planning ($4,000)

- Create Project Team
- Determine Budget
- Determine Scope
    - Project assessment/analysis
    - Analyze and verify scope of project
- Determine schedule
- Risk evaluation
    - Identify risks
    - Risk assessment
- Develop Project Management Plan
    - Define deliverables
    - Get input for project management plan
    - Designate roles
    - Verify project management plan
- Gather requirements
    - Conduct interviews, questionnaires & surveys
    - Get functional and user requirements
    - Create use case
    - Create user stories
    - Verify use cases/stories
- Technical Specifications
    - Determine programming language
    - Determining Tech Stack
    - Review Tech specifications with team
- Documentation
    - Write requirements documentation
    - Verify requirements documentation
    - Approval of documentation
- Design

- Create storyboard and website wireframe
- Front End
    - Design overall layout/base boilerplate (CSS/HTML)
    - Design Graphical User Interface (CSS/HTML)
    - Get design approval
    - Finalize design
- Back End/Database
    - Design entity relation model
    - Normalize tables
    - Get database design approval
    - Finalized database design

Executing ($22,000)

- Set up
    - Find a hosting server
    - Register Domain name
- Development
    - Front End
        - Code pages
        - Test page functionality
        - Review functionality/design according to requirements
    - Back End
        - Identify relationships
        - Create database tables
        - Review database design
        - Test relationships
        - Import data
        - Test database functionality
- Quality Tests
    - Front End
        - Verify Design and functionality
        - Verify User stories
    - Back End
        - Verify integration with Front End
        - Verify functionality and design
- Functional Tests
    - Establish a system test environment
    - Conduct Performance Tests
    - Conduct Load Tests
    - Conduct Stress Tests
    - Conduct client-server Tests
- Refine code
- Final testing

- Deployment
- Support
    - User Training
    - Project Documentation
    - Project enhancement

Monitor and Controlling ($6000)

Monitoring

- Run Periodic tests on website
- Verify system performance and functionality
- Initial Operation and Maintenance
- Implement updates

Controlling

- Customer Support system/ Q&A

Close ($2000)

- Verify all project objectives
- Project analysis
- Project review
- Sign Project off

Requirements

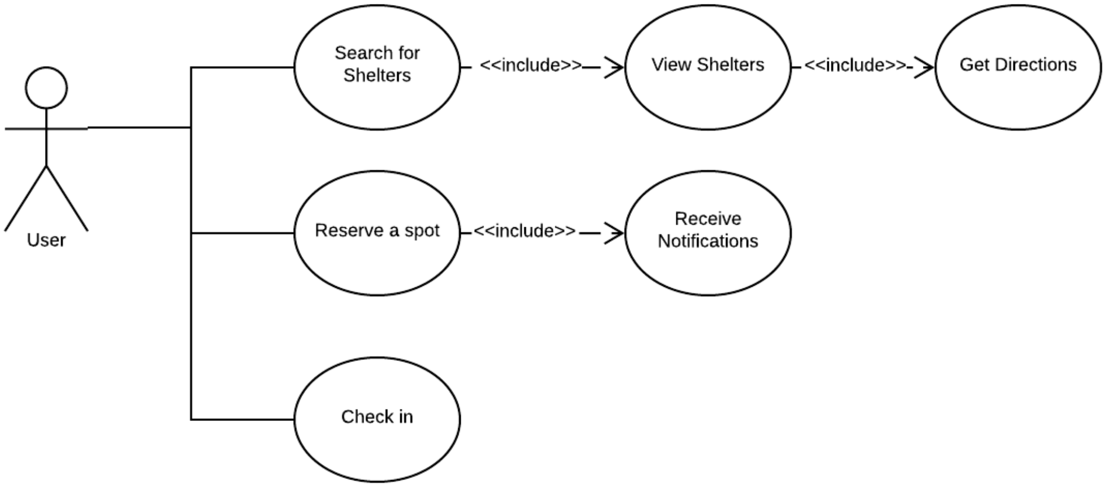
FUNCTIONAL REQUIREMENTS

a. User can look/search for shelters
b. User can register for a spot in the shelter of their choice
c. User can receive notifications of the status of the registered spot #{check db if full/return msg if full/empty}
d. User can check into the shelter through our website if they're already at the shelter
e. Database should update capacity of shelter real time or near real time
f. System should store and return location of shelters
g. System should provide directions to nearest shelter based on user location
h. System should be able to determine user location

NONFUNCTIONAL REQUIREMENTS

a. System should accommodate large amount of user (Ex: 200 users)
b. A search result of shelters should take less than 15 seconds to load
c. All database information should be up to date (furthest is one month out)
d. The database must be able to recover or restored to its data of no more than 1 hour before system failure occurred
e. All user support should be answered through chat or email no less than 15 mins (Customer Support)

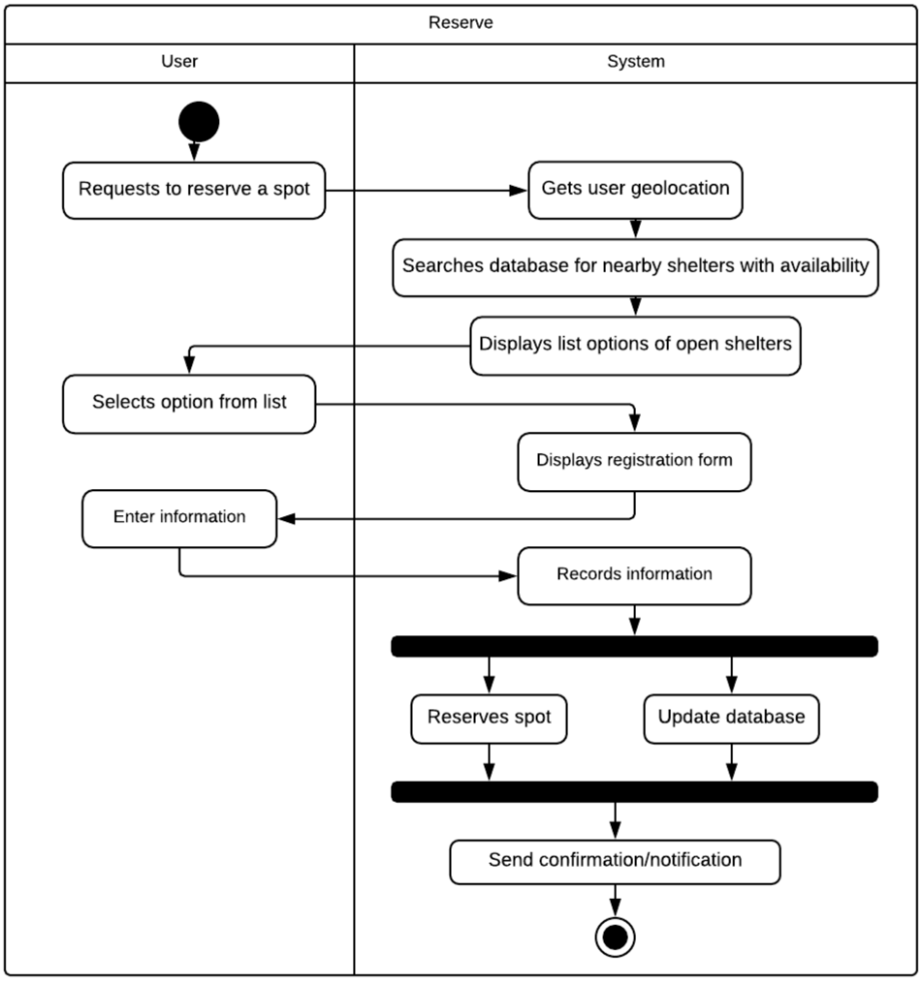
Use Case Diagram



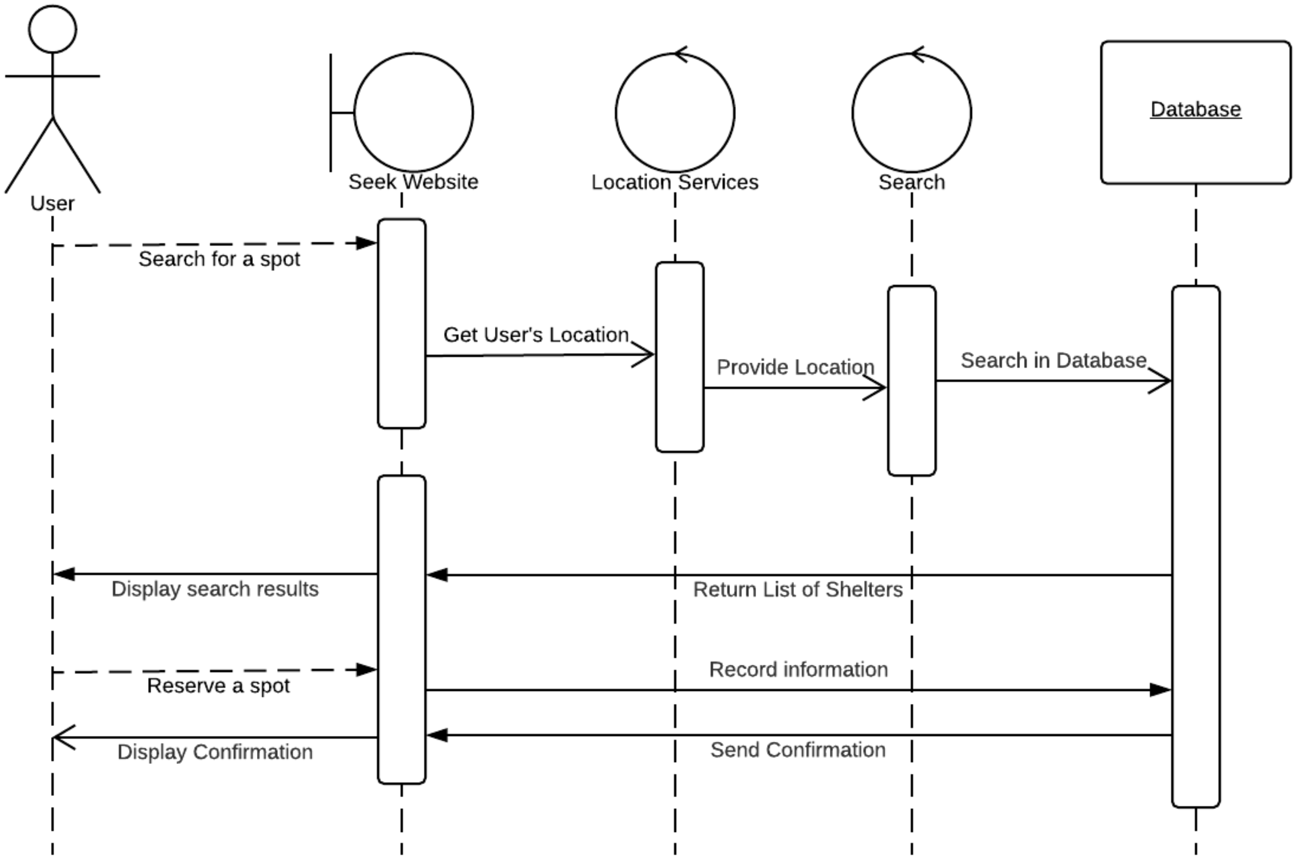
Req. ID	Requirement Description	Functional/ Non-Functional	Use Case ID	Use Case	Priority	Complexity
R1	Database should update capacity of shelter in near real time	Functional	UC1	Update	High	Moderate
R2	system should store and return location of shelters	Functional	UC2	Stored Location	High	Simple
R3	User should be able to get directions to shelter	Functional	UC3	Directions	High	Moderate
R4	User can reserve a spot in the shelter of their choice	Functional	UC4	Reserve	High	Moderate

R5	User should be able to look for a shelter	Functional	UC5	Seek Shelter	High	Moderate
R6	System should link to Google Maps to display shelter locations	Non-Functional	UC6	Maps	Medium	Moderate
R7	User can receive notifications of the status of the registered spot	Functional	UC7	Notification	Medium	Moderate
R8	System should be able to handle large number of users	Nonfunctional	UC8	Load Bearing	Medium	Complex
R9	Database info should be current (1 month)	Nonfunctional	UC9	Up-to-date	High	Simple
R10	The database must be able to recover or restored to its data of no more than 1 hour before system failure occurred	Nonfunctional	UC10	Recover	Low	Complex
R11	Website should be mobile-friendly	Nonfunctional	UC11	Mobile	Medium	Simple
R12	User can check in at the shelter through our website	Functional	UC12		Medium	Moderate

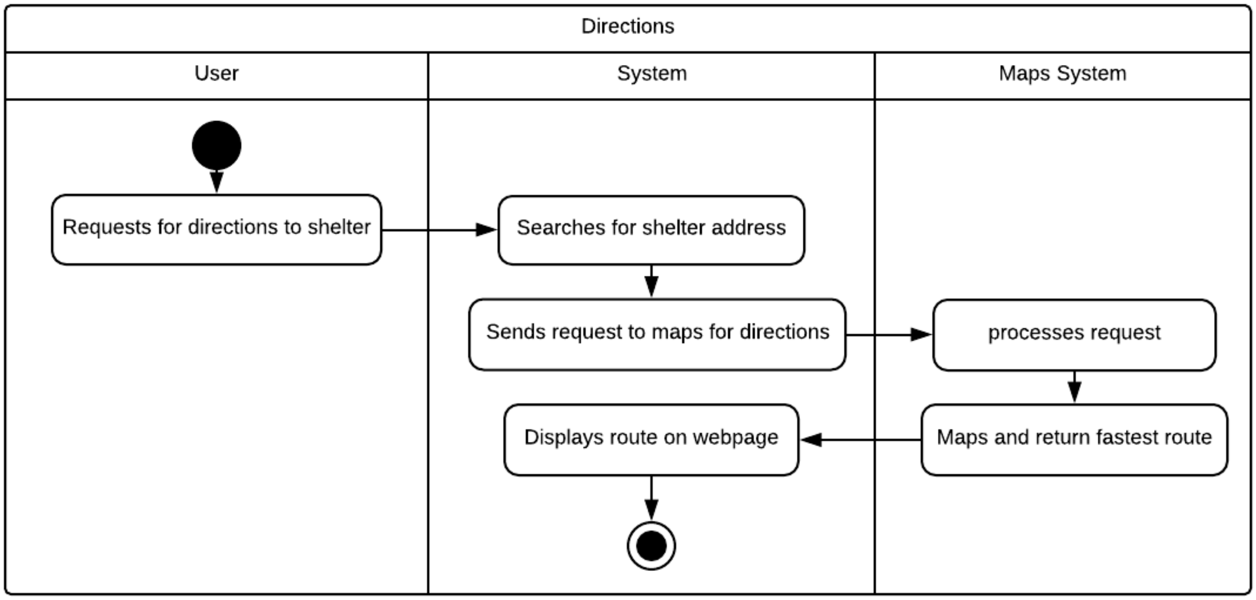
First Activity Diagram


Reserve
User
System
Requests to reserve a spot
Gets user geolocation
Searches database for nearby shelters with availability
Displays list options of open shelters
Selects option from list
Displays registration form
Enter information
Records information
Reserves spot
Update database
Send confirmation/notification

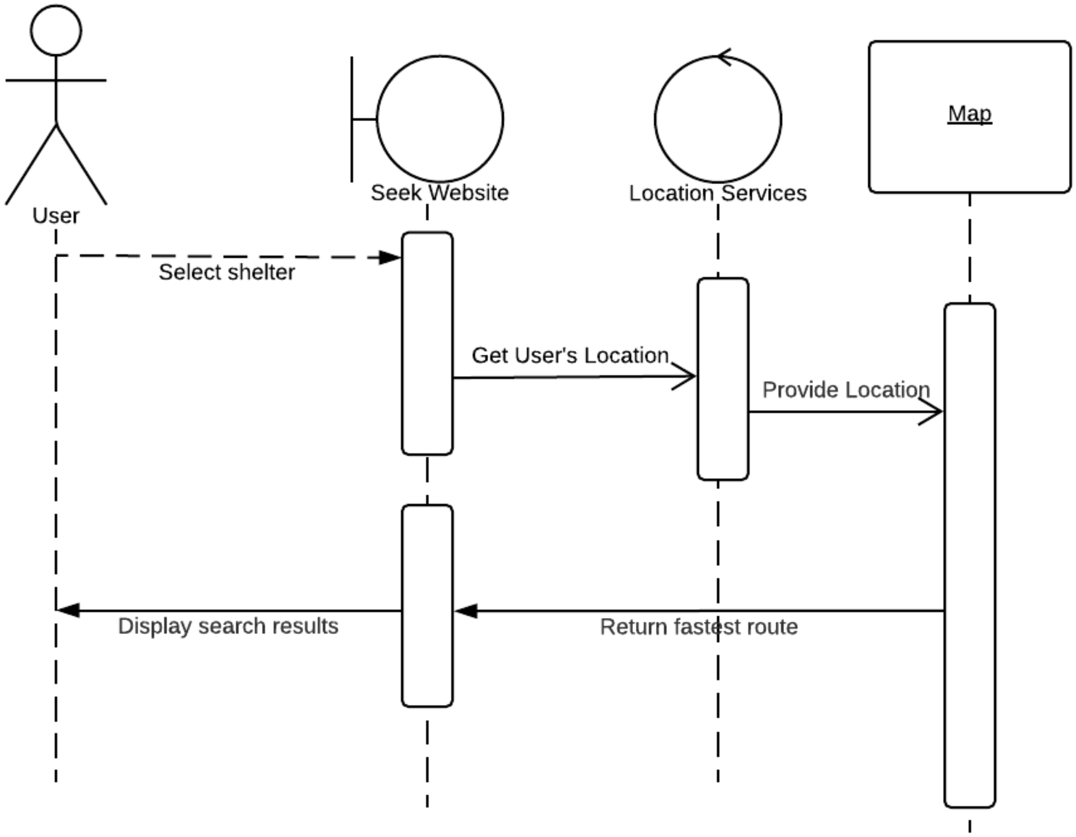
First Sequence Diagram



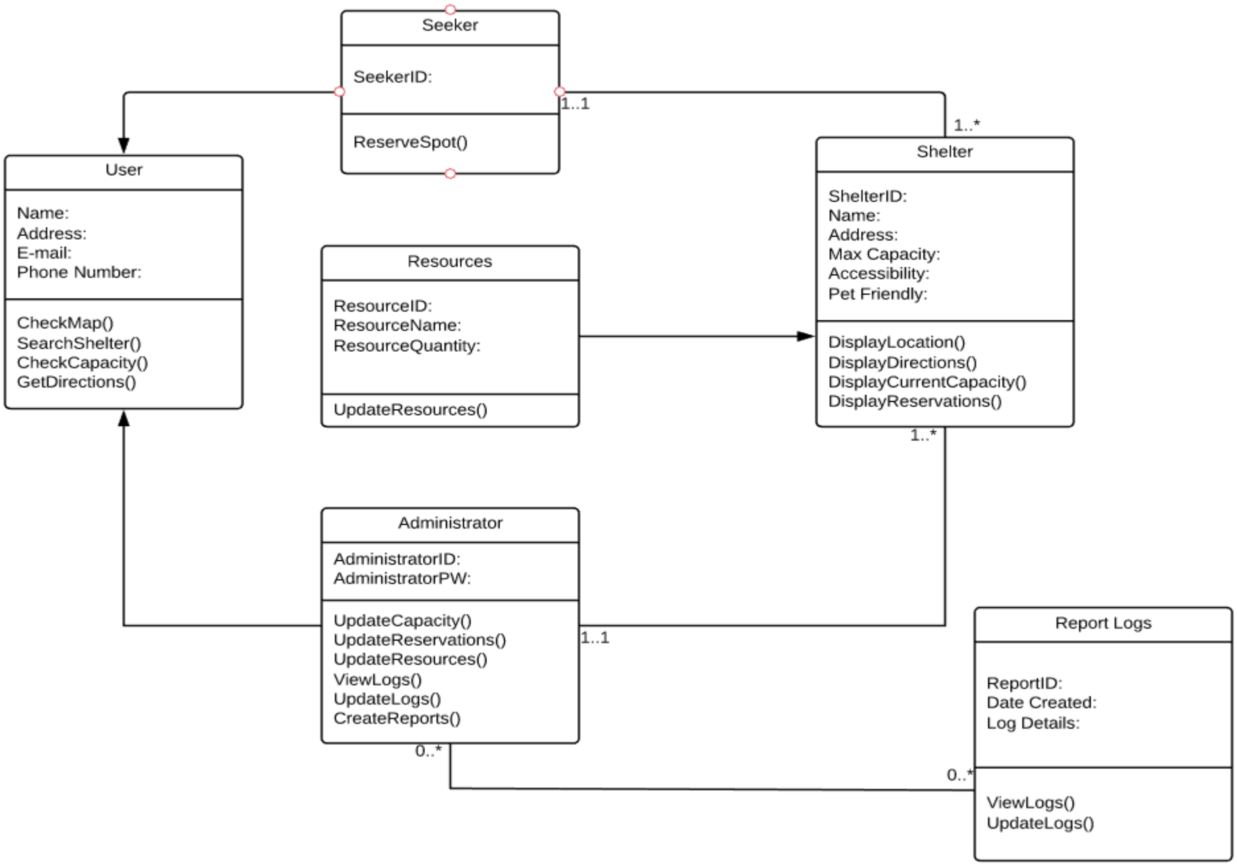
Second Activity Diagram


Directions
User
System
Maps System
Requests for directions to shelter
Searches for shelter address
Sends request to maps for directions
processes request
Maps and return fastest route
Displays route on webpage

Second Sequence Diagram



Class Diagram



Database Normalization

Logical Model

User: UserID(PK), FirstName, LastName, PhoneNum, Email, FamilySize, ShelterID(FK)

Shelter: ShelterID(PK), Name, Address, CurrentOccupants, Capacity, Borough, ZipCode, PhoneNum, Email

Inventory: ItemID(PK), ItemType, Quantity, ShelterID(FK)

1NF

Inventory

FD1: ItemID -> ItemType

FD2: ItemID, ShelterID -> ItemType, Quantity

Item (ItemID(PK), ItemType)

Inventory (ItemID(PK)(FK), ShelterID(FK), Quantity)

User

FD1: UserID -> FirstName, LastName

FD2: FirstName, LastName, PhoneNum -> Email, FamilySize, ShelterID

User (UserID(PK), FirstName, LastName)

User_Contact (UserID(PK)(FK), PhoneNum, Email)

User_Shelter_History(UserID(PK)(FK), ShelterID(FK), FamilySize)

Shelter (ShelterID(PK), Name, CurrentOccupants, Capacity)

Shelter_Contact (ShelterID(PK)(FK), PhoneNum, Email)

Shelter_Location (ShelterID(PK)(FK), Address, Borough, ZipCode)

FINAL DB ER MODEL (In 3NF)

User (UserID(PK), FirstName, LastName)

User_Contact (UserID(PK)(FK), PhoneNum, Email)

User_Shelter_History(UserID(PK)(FK), ShelterID(FK), FamilySize)

Shelter (ShelterID(PK), Name, CurrentOccupants, Capacity)

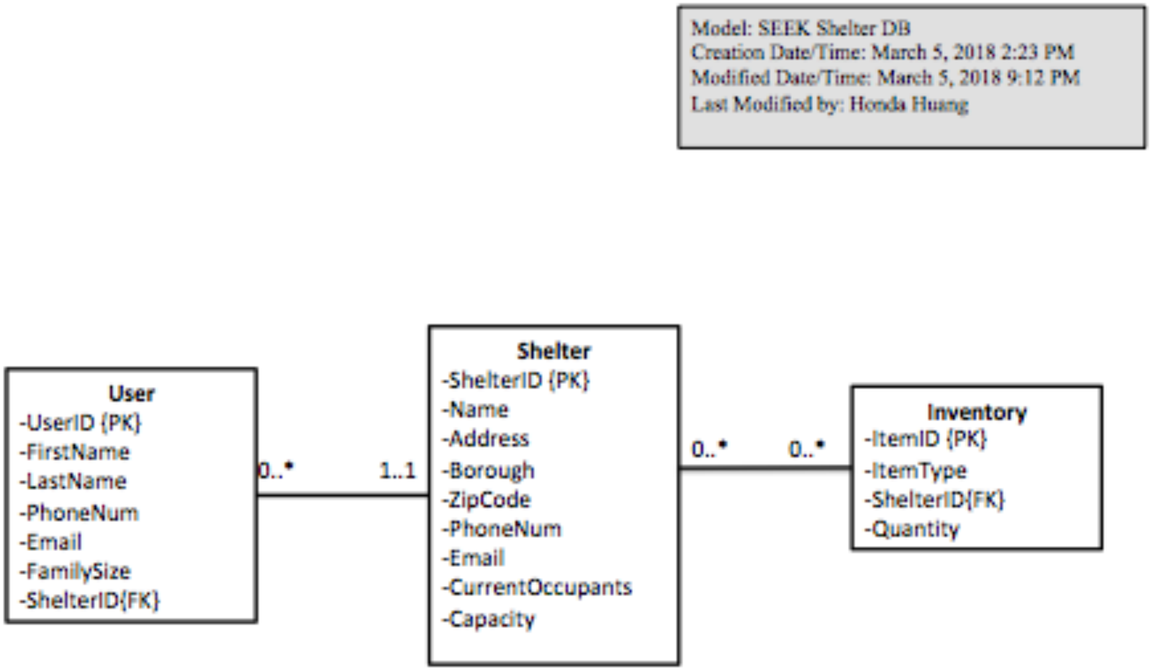
Shelter_Contact (ShelterID(PK)(FK), PhoneNum, Email)

Shelter_Location (ShelterID(PK)(FK), Address, Borough, ZipCode)

Item (ItemID(PK), ItemType)

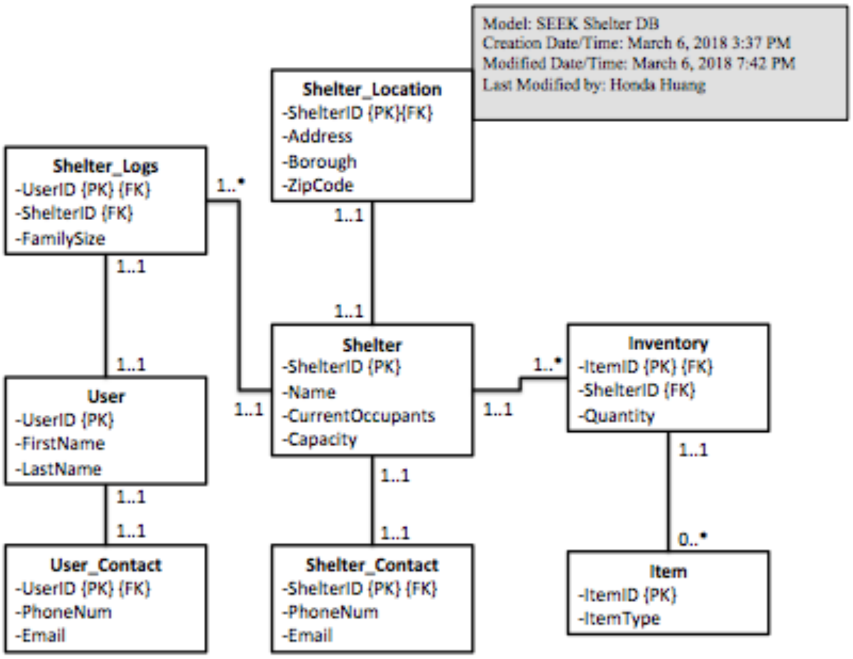
Inventory (ItemID(PK)(FK), ShelterID(FK), Quantity)

Pre-normalized ER Model



Model: SEEK Shelter DB
Creation Date/Time: March 5, 2018 2:23 PM
Modified Date/Time: March 5, 2018 9:12 PM
Last Modified by: Honda Huang

Normalized Relational Model



Data Dictionary

Shelter Table				
Field Name	**Field Length**	**Data Type**	**P/F**	**Description**
ShelterID	10	Integer	PK	Unique ID for all shelters
Name	40	VarChar		Name of shelter
Capacity	10	Integer		Maximum number of occupants that a shelter can hold
CurrentOccupants	5	Integer		Current number of occupants residing in shelter

Shelter_Location Table				
Field Name	**Field**	**Data Type**	**P/F**	**Description**

	Length			
ShelterID	10	Integer	P/FK	Unique ID for all shelters
Address	30	VarChar		Address of shelter
Borough	10	VarChar		The borough where the shelter is located
ZipCode	5	Integer		Zip code of the shelter

Shelter_Contact Table				
Field Name	**Field Length**	**Data Type**	**P/F**	**Description**
ShelterID	10	Integer	P/FK	Unique ID for all shelters
PhoneNum	15	Integer		Listed phone number of shelter
Email	30	VarChar		Contact email of shelter

User Table				
Field Name	**Field Length**	**Data Type**	**P/F**	**Description**
UserID	10	Integer	PK	Unique ID for all users
FirstName	20	VarChar		First name of registered user
LastName	20	VarChar		Last name of registered user

User_Contact Table				
Field Name	**Field Length**	**Data Type**	**P/F**	**Description**
UserID	10	Integer	P/FK	Unique ID for all users

PhoneNum	15	Integer		Listed phone number of user
Email	30	VarChar		Contact email of user

User_Shelter_History Table

Field Name	Field Length	Data Type	P/F	Description
UserID	10	Integer	P/FK	Unique ID for all users
ShelterID	10	Integer	FK	Unique ID for all shelters
FamilySize	2	Integer		Number of members in the user's family

Inventory Table

Field Name	Field Length	Data Type	P/F	Description
ItemID	10	Integer	P/FK	Unique ID for all items
ShelterID	10	Integer	FK	Unique ID for all shelters
Quantity	5	Integer		Listed quantity of available items stored in shelters

Item Table

Field Name	Field Length	Data Type	P/F	Description
ItemID	10	Integer	PK	Unique ID for all items
ItemType	30	VarChar		Name of item

Test Case

Test ID	Test Name	Test Description
TC1	Reserve	Users can reserve for a spot and are entered into the database.
TC2	Accurate Directions	Appropriate directions are displayed
TC3	Responsive	Site should scale to various screen sizes
TC4	Search Shelter	User should be able to search for nearby shelters
TC5	Received Notification	System should successfully send the User notifications
TC6	System Updates Database	Update Database Use Case
TC7	System Determines user's location	Determine User Location Use Case
TC8	System stores location	All shelter locations show on map
TC9	User Checks into the Shelter	Check In Use Case

Sprint Backlog

USER STORIES

1) As a user I want to be able to check in to a shelter
2) As a user I want to reserve a spot for a shelter
3) As a user I want to search for shelters
4) As a user I want to view shelter locations

TASKS

Sprint Backlog		Story Points	Effort	Responsible Members
User Story 1		**3**		
Task 1	Create website wireframe and Design		**3 days**	June
Task 2	Create/design page for check-in		**1 day**	June
Task 3	Create form for check-in		**1 hr**	Honda
User Story 2		**3**		
Task 1	Design and Create home page		**2 days**	June, Dewitt
Task 2	Create register form using HTML/CSS		**.5 hr**	June
Task 3	Add menu to toggle options		**.5 hr**	Mark
User Story 3		**1**		
Task 1	Create search form/bar		**.5 hr**	Mark
User Story 4		**4**		
Task 1	Create/show list of shelters		**1 day**	Honda
Task 2	Create frame and load maps API		**2 hrs**	Dewitt
Task 3	Load shelter locations on to map		**1 hr**	Justin

List of working functionality and features in Sprint-1:

- Check-in Form
- Map with all the disaster/emergency hurricane shelter locations in NYC
- Location Enabled
- Working website via Heroku
- Contact and About Page

Requirements Traceability Matrix

Req. ID	Requirement Description	Functionality	Use Case ID	Use Case	Priority	Complexity	Requirements Met
R1	Database should update capacity of shelter in near real time	Functional	UC1	Update	High	Moderate	Not yet, as we have not fully implemented the DB onto the site
R2	System should store and return location of shelters	Functional	UC2	Stored Location	High	Simple	Yes, the location of the shelters are stored onto the site
R3	User should be able to get directions to shelter	Functional	UC3	Directions	High	Moderate	Yes, the user is able to click on the pin of a specific shelter to view the directions through google maps
R4	User can reserve a spot in the the shelter of their choice	Functional	UC4	Reserve	High	Moderate	Partial, the form submission page is running but the integration with the database has not been completed
R5	User should be able to look for a shelter	Functional	UC5	Seek Shelter	High	Moderate	Yes, users are able to view the various NYC shelters on the google maps interface on the home page
R6	System should link to Google Maps to display shelter locations	Non Functional	UC6	Maps	Medium	Moderate	Yes, google maps is linked to the homepage of the site
R7	System should be able to handle large amount of users	Nonfunctional	UC8	Load Bearing	Medium	Complex	TBD, not yet tested

R8	Database info should be current (1 month)	Nonfunctional	UC9	Up-to-date	High	Simple	Yes, all shelters in NYC are included in the Database, all current users are in the Database
R9	The database must be able to recover or restore to its data of no more than 1 hour before system failure occurred	Nonfunctional	UC10	Recover	Low	Complex	TBD, not yet tested
R10	Website should be mobile-friendly	Nonfunctional	UC11	Mobile	Medium	Simple	Partial, site is functional on mobile but not entirely mobile-friendly yet

Test Case

Test ID	Test Name	Test Description	Pass/Fail/ Not Tested
TC1	Reserve	Users can reserve for a spot and are entered into the database.	Not Tested
TC2	Accurate Directions	Appropriate directions are displayed	Not Tested
TC3	Responsive	Site should scale to various screen sizes	Pass
TC4	Search Shelter	User should be able to search for nearby shelters	Not Tested
TC5	Received Notification	System should successfully send the User notifications	Not Tested
TC6	System Updates Database	Update Database Use Case	Not Tested
TC7	System Determines user's location	Determine User Location Use Case	Pass
TC8	System stores location	All shelter locations show on map	Pass
TC9	User Checks into the Shelter	Check In Use Case	Pass

Sprint 2 Backlog

Requirements	Timeline	Budget	Release
Database should update capacity of shelter in near real time	4/16/18 - 4/25/18	$4000	4/25/18
User can reserve a spot in the the shelter of their choice	4/16/18 - 4/25/18	$2000	4/25/18
User can search for shelter and the directions to get there	4/16/18 - 4/25/18	$0	4/25/18
Complete check-in form	4/16/18 - 4/25/18	$500	4/25/18
Admin login page for backend database	4/16/18 - 4/25/18	$200	4/25/18

Updated List of working functionality & features for Sprint-2:

- Check-in form that delivers user an email confirmation of the reserved spot to their inbox; Check-in form also automatically registers the user into our database and updates the database at real time
- Map with all the disaster/emergency hurricane shelter locations in NYC
- Search Function incorporated, user is able to get directions to the shelters
- Database is up and running, user is able to view the capacity and resources of shelters
- Working contact and about page
- Responsive design

Requirements Traceability Matrix

Req. ID	Requirement Description	Functionality	Use Case ID	Use Case	Priority	Complexity	Requirements Met
R1	Database should update capacity of shelter in near real time	Functional	UC1	Update	High	Moderate	Yes, the database contains and updates the current number of occupants in each shelter when information is pushed or pulled
R2	System should store and return location of shelters	Functional	UC2	Stored Location	High	Simple	Yes, the location of the shelters are stored and displayed on the site
R3	User should be able to get directions to shelter	Functional	UC3	Directions	High	Moderate	Yes, the user is able to click on the pin of a specific shelter to view the directions through google maps
R4	User can reserve a spot in a shelter of their choice	Functional	UC4	Reserve	High	Moderate	Yes, the form submission page is running and the integration with the database has been completed
R5	User should be able to look for a shelter	Functional	UC5	Seek Shelter	High	Moderate	Yes, users are able to view the various NYC shelters on the google maps interface using the search bar or the interface itself
R6	System should link to Google Maps to display shelter locations	Non Functional	UC6	Maps	Medium	Moderate	Yes ,the google maps api is linked to the homepage of the site
R7	System should be able to handle large amount of users	Nonfunctional	UC8	Load Bearing	Medium	Complex	TBD, not yet tested

R8	Database info should be current (1 month)	Nonfunctional	UC9	Up-to-date	High	Simple	Yes, all shelters in NYC are included in the Database, all current users are in the Database
R9	The database must be able to recover or restore to its data of no more than 1 hour before system failure occurred	Nonfunctional	UC10	Recover	Low	Complex	TBD, not yet tested
R10	Website should be mobile-friendly	Nonfunctional	UC11	Mobile	Medium	Simple	Yes, site is functional on mobile and has different CSS for different media screens

Test Case

Test ID	Test Name	Test Description	Pass/Fail/ Not Tested
TC1	Reserve	Users can reserve for a spot and are entered into the database.	Pass
TC2	Accurate Directions	Appropriate directions are displayed	Pass
TC3	Responsive	Site should scale to various screen sizes	Pass
TC4	Search Shelter	User should be able to search for nearby shelters	Pass
TC5	Received Notification	System should successfully send the User notifications	Pass
TC6	System Updates Database	Update Database Use Case	Pass
TC7	System Determines user's location	Determine User Location Use Case	Pass
TC8	System stores location	All shelter locations show on map	Pass
TC9	User Checks into the Shelter	Check In Use Case	Pass