(last revision January 29th, 2015)

QuAck

Ashish Singh, Ekaterina Bevinova, Nadia Vedeneyeva, Xi Wei Yin, Xiaomeng Xu

February 11, 2019

INTRODUCTION (Nadia)

Welcome to QuAck -- a social network project aiming to help students through their college journey. The idea behind QuAck is to create a unified community within a university, bringing current students and the alumni together. Unlike other social networking projects, we are eliminating an option of friends, followers, etc. We don't believe in dividing our community into subgroups of cliques or turning it into a popularity contest. We want our users to feel comfortable making new connections and approaching new people. To make it easier for users to return to a particular contact or topic he/she is interested in we've created a 'mark it' option. Much like a "subscription" or a "bookmark" it will allow our users to track their interests. However, the "marks" will be hidden from public allowing for a judgment free environment. We plan to keep university communities isolated from one another in order to form a stronger bond within. In order to get a QuAck account, users are required to have a 'stevens.edu' email address. If the project reaches its success criteria, we plan to expand our services to other institutions.

Once a student is registered with QuAck and has created a profile, he/she is ready to make new connections, seek help or answers to college related questions. The home page consists of three categories of service provided by QuAck. Our application will provide a marketplace, a job market and a social place. In the marketplace students will be able to look for roommates, housing and carpool pals. Also they will be able to buy and sell each other books, furniture, appliances and other products necessary to settle in a new environment. The job market section will provide students with work opportunities by offering job and internship postings. We are hoping to create a community where after graduation the alumni will return to QuAck and help current students on the job market. QuAck is meant to help students have some fun too. That is why we are creating a social place. Here our users will be able to find information about events happening in the community, find people with similar interests, or simply ask if anybody is interested in seeing a movie or going to a concert. Additionally, social place will have a 'Q and A' rubric, where students will be able to ask their peers questions and seek advice.

Starting a new school is not easy. Our goal is to help students help each other. So let's get QuAcking!!!

ROLES AND RESPONSIBILITIES (required)

These vary for each type of product and, for small projects, folks may serve multiple roles. This is a list of common roles we have used for software development:

Development Lead (Raymond)
Buildmeister (Raymond)
Architect (Raymond)
Developers (ALL)
Test Lead (Katya)
Testers (ALL)
Documentation (ALL)
Documentation Editor (Raymond)
Designer (Nadia)
User advocate (Katya)
Risk Management (Nadia)
System Administrator (Raymond)
Modification Request Board (1 leader, multiple representatives)
Requirements Resource (Katya)
Customer Representative (All)
Customer responsible for acceptance testing (All)

METHOD (required)

These are unique to software development, although there may be some overlap.

- Software:
  - Language(s): Python 3.5; Java 11; Swift 4.2
  - Operating System(s): macOS Mojave 10; Windows 10
  - Software packages/libraries used with release/version number: Xcode 9; Android Studio; Python standard libraries, Jinja2, Java class libraries.(Not very sure about this one)
  - Code conventions :
    - Python - https://www.python.org/dev/peps/pep-0008/
    - Java: https://www.oracle.com/technetwork/java/codeconvtoc-136057.html
- Hardware:
  - Development Hardware: Macbook, Windows-book
  - Test Hardware: Macbook, Windows-book, IPhone, Android Phone
  - Target/Deployment Hardware: IPhone, Android Phone, PC
- Back up plan (individual and project): Weekly upload to Github
- Review Process:
  - Review targets: Requirement; Architecture; Design; Code; Quality
  - Formal approach: Requirement, Design, Quality. Informal: Architecture; Code
  - Whole team would be responsible for reviewing and resolving any issues.
  - Code readings? - TBD
- Build Plan:
  - Revision control system and repository used - Github(raymondyin/sit-ssw690-group-work)
  - Regularity of the builds – weekly
  - Deadlines for the builds – Each Sunday night by 11:30 PM
  - Multiplicity of builds - TBD

    - Regression test process – see test plan
- Modification Request Process:
    - MR tool - Github pull request
    - Decision process (board – if more than paragraph should point to alternate description) - Every merge needs approval from team leader plus another team member.
    - State whether there will be two process streams one during development and one after development - TBD

Virtual and Real Work Space

Virtual Space: Google doc, Github, Whatsapp
Real Space: Library, Altorfer 115

COMMUNICATION PLAN (required)

“Heartbeat” meetings

“Heartbeat meetings” will be held weekly at Wednesday afternoon from 4 to 5 PM in library. All development team members will be attending. In this meeting, development team will briefly go over opening issues and difficulties. Any issues as well as solutions should be recoded.

Status meetings

Status meeting will be held biweekly at Thursday afternoon from 4 to 4:30 through google hangout. This meeting will be focusing on progress tracking, arrangement of deliverables for next sprint, and process review. All team members should be attending.

Issues meetings

If any team member needs assistance, he/she shall send out request of an Issue meeting to other team members. This meeting should be scheduled according to attendees’ agendas,

TIMELINE AND MILESTONES(Raymond)

This section should be crisp containing 4-10 milestones for the duration of the project, each of which would trigger a re-issuing of this document to report on progress. Each milestone should define a 100% complete item, should list the critical participants and list begin time and end time. Each time you re-issue this document you should highlight changes with italics or bold – colors will not show up on a photocopy.

Note that for this project we have a few time boxes. They are:

- Week of February 2nd – description of first demo
- Week of February 16th first demo, description of second demo
- Week of March 2nd second demo, description of third demo
- Week of March 23rd third demo, description of fourth demo
- Week of April 6th fourth demo, description of final product
- Week of April 27th final product

TESTING POLICY/PLAN (optional–software relevant)(Katya)

Once the features and the requirements are set a separate Testing Plan Document can be generated and referenced in the Development Plan Document.

Test Plan Identifier:	Master plan 1.0 Several subsequent testing plans will be generated based on different platforms that will be tested.
Introduction:	● Testing will be employed in all phases of SDLC. It will become part of the development process. ● The goal of testing is to deliver the end product with fully functioning high priority features with minimal number of bugs by implementing several known testing techniques such as gray, white, black box testing, data flow testing, unit case testing. ● A more thorough testing plan will be generated once the requirements and the features have been established with an estimated delivery time of week 4.
Test Items:	● iOS 12.1.4. ● Oreo 8.0-8.1 Linux Kernel 4.10 ● Pie 9.0 Linux Kernel versions 4.4.107, 4.9.84, 4.14.42

Features to be Tested:	Defining of a set of features is in progress. Tentative features: ● Log in/out ● Customization of the profile page ● 3 different categories available on the home page as buttons ● Ability to “bookmark” people, events, product postings
Approach:	● The overall Approach will be determined once the features and requirements are established. ● Manual Testing ● Automated Testing ● White Box/Black Box/Gray Box ● Security Testing
Item Pass/Fail Criteria:	To be determined.
Estimate:	Depending on what tools would be used and what features tested, the estimated cost of the whole testing project would not exceed $30.00 in monetary value.
Schedule:	Testing will be generated and ran alongside development process.
Testing Tools:	Possible Testing Tools to consider: Tricentis Tosca Testsuite Perfecto Mobile Monitoring Smart Software Testing Solutions pCloudy
Staffing and Training Needs:	● Specify staffing needs by role and required skills. ● Identify training that is necessary to provide those skills, if not already acquired. ●

RISKS (Nadia)

Risk	Strategy Category	Risk planning
Staff turnover	Avoidance	Our group consists of five people instead of four. In case of team member getting sick or dropping the class, our group will have the same amount of members as other groups in the class.
Management change	Contingency	All members of the group remain on top of the things and participate in decision making. Subsequently, management change would be inconvenient, but manageable.
Hardware unavailability	Minimization	For the scope of the project our laptops are the only hardware we require.
Requirements change	Minimization	We use agile software development method. By doing so we minimize the effect requirement changes might have on our project.
Specification delays	TBA	TBA
Size underestimate	Contingency	The most important features are being implemented first. If it turns out that size was underestimated we are planning to drop least important features.
CASE tool underperformance	TBA	TBA
Technology change	TBA	TBA
Product competition	Contingency	Agile development method allows us to add new, unique features in case a competitive product emerges and such are required.

ASSUMPTIONS(required)(Katya)

- Hardware Used: MacBook Pro, Android Samsung Galaxy, iPhone.
- Software Used: Trello, GitHub, Google Drive, WhatsApp, Google Hangouts.
- Core Requirements: performance, scalability, security, availability.
- Depending on everyone's schedule and the overall progress, Spring Break can be utilized to catch up or make progress on the project.
- Agile method is utilized with frequent "real time" feedback.
- Estimated cost of the overall project is $0.
- Everyone will be involved in every aspect of the development process to gain experience and exposure to different processes of the software development.
- Weekly meetings outside of classroom are mandatory. Depending on the load of the work that needs to be done more frequent meetings might have to be scheduled.
- Our team intends to deliver a working program that provides social platform for all the current students on campus. This program will be available through the web browser as well as a downloadable app that can be used on Android and iOS devices.
- Rewards: Finished Product that could be successful if implemented right and on time; good grade; course credits; eventual graduation.

DISTRIBUTION LIST(Katya)

Professor James Rowland
Professor Richard Kempinski
SSW-690-A Class

MORE OPTIONAL SECTIONS:

These should be self-explanatory.

Worry beads

I add this section to describe the things as manager I am most worried about at the time of latest document issue. This section is useful because it helps you to focus on the parts most likely to fail. Sometimes, I segment the worries by time scale: day, week, month, quarter … lifetime.

Documentation Plan

Many years ago we had much too much documentation, now we have precious little – this has to change. Write documentation as if you'll need to personally support the project forever – you just might need to and you'll be glad you took the time to document the obvious, the not so obvious and the obscure. As an example, it's useful to document alternate architectures and designs you did not pursue along with the rationale. What were the "gotchas" you were trying to avoid?

Build Plan

When builds and testing become complex, this might be a separate section or point to a separate document.

User Stories

Account Management

As a user, I should be able to create a “Quack” account linked to their Stevens email in as few steps as possible [priority: Must have]

As a user I should be able to recover or change my password [Must have]

As a user I should be able to update my profile. I should be able to:

- Add courses [Must have]
- Add internships [Must have]
- Add projects (in limited space) [Should have]
- Choose interests (Sports, music etc). Could be a list of checkboxes so it is quick and uniform [Must have]

As a user I should be able to select when types of notifications they want [Must have]

As an admin, I should be able to edit/remove any posts on the website/app [Must have]

Marketplace

As a user, I should be able to post an Ad to buy or sell or rent. I should be able to upload or remove images of the product [Must have]

A user should be able to edit, delete or disable an Ad [Must have]

As a user I should be able to search for a product I am looking for. I should be able to either post a query (which others can also read) or direct message the student [Must have]

Events/Social

As a user, I should be able to create an event. Everyone who has listed that activity as an interest in their profile should receive a notification [Must have]

As a user, I should be able to mark or remove my interest in an event. E.g. the notification includes a link to confirm interest [Must have]

As a user, I should be able to view the names of all other students who are interested in the event [Must have]

As a user, I should be able to post messages on the events page and plan how to proceed [Must have]

As a user, I should be able to chat live with people interested in an event [Could have - Won’t have this in the first release]

As a user, I should be able to delete, disable or update an event that I have created. An event may get disabled automatically after the event timings have passed [Must have]