

Skill-It

Group 4

Members: Madhav Jhawar, Daniel Bendersky,
Jeremy Hong, Braeden Turner, Ryan Franzosa

Manager: Trent Maziarz

University of
Massachusetts
Amherst

BE REVOLUTIONARY™



Description

We are building a skill-based student marketplace. The way this works is students create a profile with the fields of majors, year, and whether they're interested in working, posting gigs, or both. Students who selected posting gigs are then able to post opportunities, and students who selected that they were interested in working can pick up these opportunities by searching for them. Students are then able to add themselves to these opportunities.

There will be a messaging platform built-in that allows students to discuss with one another the details of the opportunities. Students who worked with other students on opportunities can review each other and leave feedback.

Students can then view other students' profiles and their ratings, and can search for other users who match the skills they're looking for and reach out to them proactively.

Use Case Flowchart

1

Students create their profile, indicating their interests, major(s), year, and whether they're interested in working, posting gigs, or both.

3

Students search for opportunities

4

Students add themselves to opportunities

5

Students can message each other to discuss details

6

Students review other users and leave feedback

2

Students post opportunities

8

Students can search for other users who match the skills they're looking for and reach out to them proactively

7

Students can view other users profiles and ratings

1. Creating User Profile

The UMass Student enters the platform and selects ‘Sign Up’. The System directs the Student to the registration form and prompts the Student to enter their profile information. The Student can apply any number of predetermined skills, or “tags” to highlight on their profile. Once done with the registration, the Student selects ‘Create Profile’ and the System validates and stores a new record that can be used for future login.

1: Creating User Profile cont.

| | |
|------------------------------------|---|
| Goal | UMass Student creates a user profile |
| Primary Actor | UMass Student |
| Precondition | Student is not logged in and is directed to login screen |
| Success End Condition | A new record is created for the Student and an active session is started |
| Failure End Condition | No user data is saved, Student remains on sign-up page with error feedback |
| Trigger | Student selects 'Sign Up' on login screen |
| Main Success Scenario | <ol style="list-style-type: none">1. Student opens platform2. Student is directed to login screen and selects 'Sign Up'3. System redirects Student to registration form4. Student enters required information and selects 'Create Profile'5. System stores entered data and a new student profile is created.6. System redirects Student to main dashboard |
| Variations (Error Scenarios) | <p>3a. Student entered invalid information (no '@' in email)/ left required fields blank</p> <p>3b. Student record is already present and the system offers link redirecting Student to login screen.</p> |
| Variations (Alternative Scenarios) | <p>4a. Student decides to not complete registration and selects 'Cancel'</p> |

2. Posting an opportunity

Once logged in, a student is able to post about new opportunities which they are looking for help with. The student will select a name, category, and optional image for the post. They can also add a description and select from the set of predefined skills they're looking for (tags). Once they hit post, the opportunity will show up in relevant searches on the website and in the user's profile page.

2: Posting an Opportunity - Dan

| | |
|------------------------------------|---|
| Goal | Students post opportunities |
| Primary Actor | UMass student looking for help |
| Precondition | Student must have an account created on the platform and must be verified as a student |
| Success End Condition | Opportunity can appear in student's searches |
| Failure End Condition | Student is notified that their event is not valid |
| Trigger | When a student is looking for help on a project, gig, or startup |
| Main Success Scenario | <ol style="list-style-type: none">1. Student writes a name for the opportunity2. Student selects a category for the opportunity3. (Optional) Student may upload an image, add a description, and select relevant skills for the opportunity.4. Student hits the post button.5. Opportunity is stored in system data and appears in relevant searches and on the user's profile. |
| Variations (Error Scenarios) | -Student fails to enter a name or category -Student forgets to hit post button |
| Variations (Alternative Scenarios) | -Student comes back to edit the post later on -Category of opportunity does not match any categories available |

3. Searching/filtering opportunities and people

A student accesses the opportunities page, and wants to search for specific opportunities for work. They can search for keywords, or apply filters based on the job description of predetermined “tags” (major, time requirement, skills, etc.). The user can also search or filter people who are explicitly open for opportunities to reach out to them directly.

3: Searching/filtering opportunities and people- Braeden

| | |
|------------------------------------|--|
| Goal | Search/filtering options for work opportunities and profiles |
| Primary Actor | UMass student |
| Precondition | UMass student has logged on. User is interested in specific work opportunities or finding other students open to work. |
| Success End Condition | Relevant work opportunities or students appear on UI. |
| Failure End Condition | No relevant work opportunities or students appear after search/filter. |
| Trigger | User types in search bar or applies specific filters |
| Main Success Scenario | <ol style="list-style-type: none">1. User logs onto website2. User goes to work opportunities page or profiles page3. User types in search bar for specific opportunities/people or applies filters4. Relevant opportunities/people appear that a user can click on for more info |
| Variations (Error Scenarios) | Search/filtering system does not work. By default, all work opportunities/people should appear on page through infinite scroll, so user can see all opportunities regardless. |
| Variations (Alternative Scenarios) | N/A |

4: Students add themselves to opportunities

A student searches for gig opportunities. Student “signs-up” for the gig. The page will alert the student that a notification was sent to the gig requestor and a message channel between the student and the requestor is opened for questions. Student will wait for requestor to either message them directly for questions and/or accept/reject the student for the gig. Student may revisit the site and log-in and see the message channel and status of the offer.

5. Messaging between Students

After a student accepts an opportunity, a messaging thread is established. The student can then open that messaging thread and begin to say any message, which is usually going to be clarifying questions before they start their work. After sending a message, the student who posted the opportunity receives the message and has the opportunity to reply back to it. After the “student employer” replies back to the student, the student may ask more questions. This messaging process repeats until the student who picked up the opportunity is ready to begin their work. The messaging thread will be open during the duration of the project.

5: Messaging between Students - Madhav

| | |
|------------------------------------|---|
| Goal | Messaging and asking questions before a student begins their work |
| Primary Actor | UMass student(s) |
| Precondition | The student is logged in and found a help request they are interested in |
| Success End Condition | A messaging thread is established where students can ask clarifying questions about the help request |
| Failure End Condition | The student is not able to message the “student employer.” |
| Trigger | The student presses “message” on the job posting, which will have the “student employers’ conditions” attached |
| Main Success Scenario | <ol style="list-style-type: none">1. The student logs onto the website2. The student finds a help request3. The student accepts the request4. The student presses their “student employer” and messages them; a messaging thread is then established |
| Variations (Error Scenarios) | <ol style="list-style-type: none">1. The “student employer” doesn’t receive the message; the message doesn’t go through2. Once the student employer receives the message, they’re not able to reach the student messenger |
| Variations (Alternative Scenarios) | <ol style="list-style-type: none">1. The student is able to edit or modify their message |

6: Leaving Feedback and Rating

Once a student has completed a gig posted on the site and both the worker and requestor agreed the job has been completed, the site gives an opportunity to give feedback to both the worker and requestor. Either party clicks on link and is forwarded to a form to leave a rating and feedback. If the requester requested a group of students, requester has an opportunity to either holistically or individually evaluate them.

After completing the form, the party clicks submit and the system stores the form and appends a compacted version to the other party's public profile.

6: Leaving Feedback and Rating

| | |
|-----------------------|---|
| Goal | Student leaves feedback and rating on other students after a job. |
| Actors | Students. Two student parties: one gig requestor and one gig seeker. |
| Precondition | Student logged on application and the gig posted has been completed |
| Success End Condition | Student successfully left a rating and accompanying feedback on the other student party |
| Failure End Condition | Student failed to leave a rating and feedback on the other student party |
| Secondary Actors | The opposing party: either the requestor or job seeker |
| Trigger | Gig has been completed (agreed by both parties) and either party wants give feedback. |

6: Leaving Feedback and Rating cont.

| | |
|------------------------------------|--|
| Main Success Scenario | <ol style="list-style-type: none">1. Student logs in to account2. System lists completed gigs and an opportunity to leave a feedback on the other party3. Student clicks on one of those links4. System redirects student to a feedback page5. Student lists rating and accompanying feedback6. Student submits7. System stores and post the review underneath student's profile |
| Variations (Error Scenarios) | <ol style="list-style-type: none">1a. Student fails to log in via wrong user or password3a. student declines to give feedback4a. System fails to load in link due to network or internal error5a. Student fails to fill form our correctly6a. Student presses cancel7a. System fails to store and post the review due to network or internal error |
| Variations (Alternative Scenarios) | <ol style="list-style-type: none">1. Student is already logged in to account2. Student and requestor both mark gig as done3. Notifies both party an opportunity to leave a quick feedback on the other party4. steps 3-7 |

7. Viewing user profiles

Student searches for a list of profiles or gig opportunities. Student clicks on either the requestor profile directly or the requestor profile listed in the gig. System redirects user to a page that describes the profile. The profile includes the requestor's name, year, major, previous gigs posted/worked on, and a compact slide show of feedback and ratings.

8. Creating new Tags

When applying tags to profiles and opportunities, Students can choose to create a new tag. In the dropdown of predefined tags, the Student can choose “Create new tag” to better reflect skills not present in the current list. The Student will type the new tag name in a textbox and the System will validate for uniqueness and store the new tag. The tag will now be present for all Students creating profiles, searching, or posting opportunities.

The End

University of
Massachusetts
Amherst

BE REVOLUTIONARY™

