

ECE5830- Software Engineering Project



G003 - Project Gremester

Avanti Deshmukh, Harsha Pitawela, Linh Pham, Julia Chaloupka

Abstract

Currently, there is no shared web-based platform available for collaboration between prospective graduate students and university faculties during the grad school admission process. Students have to upload their application documents separately to each university's web-based system. Faculty members may or may not have a web-based system to evaluate student's profiles and record their evaluation ratings for next steps in admission process. Given these issues, the goal of this project is to develop a web-based system which can provide a collaboration platform to streamline the workflow of the admission process between students and faculties.

Gremester is the one-stop platform for graduate committee faculty members and potential graduate students to connect with each other. For potential graduate students, the platform provides the opportunity to build a profile with exam scores, undergraduate education details and important documents such as résumé and statement of purpose. Gremester also helps prospective students find graduate schools with better chances of admission. Moreover, the platform provides a discussion forum that allows prospective students to communicate with each other. Among the benefits offered to faculty is the ability to filter potential candidates on a range of academic criteria and evaluations of students. The team incorporated Agile methodology throughout 6 sprints over the semester and incorporated an extensive adoption of Test-Driven Development and Behavior-Driven Development practices. Gremester is implemented using the Ruby on Rails web-framework and hosted on the Heroku server.

Table of Contents

❖ Project Overview	3
❖ User Manual	6
❖ Test Plan	11
❖ Meeting Minutes	16
➤ Sprint 1	16
➤ Sprint 2	22
➤ Sprint 3	25
➤ Sprint 4	29
➤ Sprint 5	31
➤ Sprint 6	34
❖ Sprint 1-5 Overview	36
➤ Sprint 1 overview	36
➤ Sprint 2 overview	41
➤ Sprint 3 overview	45
➤ Mid-term review	48
➤ Sprint 4 overview	54
➤ Sprint 5 overview	58
❖ Appendices	62
➤ Poster Template	62
➤ Lofi Sketches	63
➤ UAT Documents	73
➤ User Guide	87

Product Statement

Currently, there is no shared platform available for collaboration between prospective graduate students and university faculties during the admission process. Students have to upload their application documents separately to each university's web-based system. Faculty members may or may not have a web application to evaluate student's profiles and record their evaluation ratings for next steps in admission process. This project aims to develop a web-based system which can provide a collaborative platform to streamline the workflow of the admission process. This system will be a one-stop platform for managing the application process for students as well as faculty members.

Project Goals

The aim for this project is to create an application for prospective graduate applicants of computer science and faculty members. This application will have the following salient features for -

Students -

- The students will be able to create their profile, add their undergraduate details, GRE/TOEFL scores and other important documents such as SOP, resume, etc.
- They will be able to search information about US universities such as university ranking & type, location & weather conditions, acceptance rate, average tuition fee per year and other information.
- The application will also provide probability score indicating chances of admission to a student for a particular university.
- Besides, it will provide a discussion forum that allows communication between students themselves.

Faculty -

- Among the benefits offered to faculty are the opportunities to filter potential candidates based on range of academic criteria such as grading scale, GRE score, research interest, MS/PhD degree objective, undergrad university
- They will be able to view and evaluate students' profiles
- The faculty will be able to view their evaluations as well as evaluations of other faculty members for a student profile.
- They will also be able to view details of student's undergraduate university such as university description, location using Google Map, ranking, acceptance rate, link to the website

Anticipated Challenges

We expect a few challenges regarding some aspects of our project. One is the significant amount of data for different US universities will be needed for some features of the application

that are developed specifically for graduate applicants for their research while selecting the universities to apply. Collecting this data from different sources would be a challenging task. Anticipated solution would be to come up with a data scraping tool to extract data from different sources. The other challenge would be to store documents uploaded into our system by students/faculty. The potential solution would be to use Google Cloud to store the documents.

Priority

Choosing the university for graduate studies requires a lot of effort. Different factors such as location, financial aspect, academics and career options are important in making decision. Particularly for students who decide to study in different countries, restriction in terms of information makes it even more difficult to make an informed decision. This application will support this requirement. Also, the faculty component of this application is proposed by the external product owner which will provide university faculty members to evaluate student profiles on a web-based system instead of manually working through spreadsheets.

Sustained Benefits

This web-based application will be a good platform to keep record of the application profile evaluations by faculty members over number of years. The system also stores application status of all users, which will be helpful in matching students' with their prospective programs. After several years of operation, the system will have a massive dataset which could be used by graduate universities to identify in which regions would they need to promote their programs. In the same manner that Glassdoor has become an FAQ platform for potential job seekers, Gremester would be an FAQ platform for potential graduate students.

Leveraging

Some commercial websites provide information to students about US universities rankings, available programs, etc. This project enhances the benefits and provide comprehensive assistance in the sense that it will provide common platform for collaboration during application process between applicants and university faculty members.

Speed to Functionality

The Project will take approximately 2 sprints to be functional and have positive impacts on the stakeholders involved. Sprint 1 will mainly involve student/faculty sign and login, while Sprint 2 will start creating student profiles and university information lookup. These features will help us to build the faculty component of the system which we are planning to work on from sprint 3 and later.

Technology

- Ruby on Rails
- Bootstrap, HTML and Javascript
- Jasmine for Javascript Testing
- Cucumber User acceptance testing
- RSpec for Unit testing

- SimpleCov and Teaspoon for test coverage
- TravisCI for continuous integration
- Heroku for web deployment
- Google Cloud Server for Maps API and document upload

User Agreement Policy

- The site will not hold any liability for the accuracy of information provided by any user registered as a student. It shall be the responsibility of the institutions that are admitting them to verify them using official records.
- The site will have the authority to remove any inappropriate or offensive ideas, questions, posts or comments posted on the site by any user.
- The site shall not be held accountable for any abuse resulting from sharing of email addresses or other contact methods.
- The site will not share a user's details with a third party without explicit permission from the user.

User Manual

Coding Style Guide

This section describes the coding style to be used for our project.

- No end of line whitespace
- No tab characters (indentation is only done with spaces)
- No more than 3 blank lines in a row.
- No more than 3 blank lines at the end of the file
- variables shall use snake case
- No unused variables in the code
- Global variables shall be all uppercase, start with GLB_ prefix

User Stories Format

As a < type of user >, I want < some goal > so that < some reason >.

Local installation for developers

To run the project locally, please clone the repository with

```
git clone git@github.com:UIOWA5830SP19/SPP300.git  
cd SPP300
```

Next, please make sure your machine has Rails installed. If not, you can install by running

```
gem install rails
```

Install all the required gems

```
bundle install --without production
```

Run database migration commands to setup database

```
rake db:migrate  
rake db:schema:load  
rake db:seed
```

After that, you can start the rails server by

```
rails server
```

If there is no error given, point your web browser to <http://localhost:3000> and see Gremester app actually running!

Local installation for end users

The app is currently hosted on <https://gremester.herokuapp.com/> Please go to the link to use or test our application. Once you go to our website, this is the landing page of Gremester



For further information in how to use the system, please refer to our User Guide document in the Appendices section. Any question or concern can be directed to anyone in our team.

Contacts

For any question or concern about the project, please contact one of the following people

- Avanti Deshmukh (avantiharish-deshmukh@uiowa.edu)
- Harsha Pitawela (harsha-pitawela@uiowa.edu)
- Linh Pham (linh-t-pham@uiowa.edu)
- Julia Chaloupka (julia-chaloupka@uiowa.edu)
- Hans Johnson (hans-johnson@uiowa.edu)

Issues Labels

Issues will be used to implement an Agile Project Management system within GitHub. Issue template will be used to keep track of bugs, enhancements, or other requests. The developer requires to fill in following information in the issue template:

1. Description of the issue
2. Input required
3. Expected and actual result
4. Additional information

All the issue need to have T[1-3], S1[1-4] and sprint label.

We decided to use ZenHub for Github issue management. Please go to our <https://bit.ly/2DDX4nv> for more information

Label Name	Label Color	Label Description
T1-Defect	#ef5f3b	This label is for problem found in production code
T2-Enhancement	#a5f6ff	This label is for improvement made to the existing core feature of the app
T3-Task	#fce6ec	This label is for a new core feature
S1-Urgent	#b60205	This label is for tasks that must be completed immediately

S2-High	#f71b3c	This label is for tasks that are important for the progression of the project
S3-Medium	#8e1dc6	This label is for tasks that need to be done not immediately
S4-Low	#14ad24	This label is for a task that do not need to complete soon
Sprint#	#4192f4	This label is specific for the sprint

Agile Board

Processes for each issue: Icebox -> Backlog -> In Progress -> Review -> Done.

- Icebox: Features which are yet to be implemented but are not of the highest priority.
- Backlog: prioritized features list, containing short descriptions of all functionality desired in the product.
- In Progress: Features in the current team which team is currently working on.
- Review: PR submitted and to be approved by the product owner.
- Done: PR merged into the master branch and the story is completed.

When is an issue done?

- Over 80% code coverage
- The new code should not break any existing test code.
- At least two other members should approve the pull request before merge.
- One of the two members approving the pull request should be the product owner for that sprint.
- Only the internal product owner can move issues to Done and close in ZenHub.

Working in parallel

During the sprint planning process, product owner and the team will identify user stories that need priorities and break down those stories into engineering tasks. Tasks are then assigned based on team capacity and order of importance. Tasks with higher complexity will require work from more than one developer and the team have to delegate the task accordingly to avoid any block from each other work.

User acceptance testing protocol

Once all the planned features for a sprint are completed, the team is required to send an email to Prof. Hans Johnson along with the User Acceptance test (UAT) manual. The UAT manual should include number of tests to be performed along with the steps, their success criteria and other details necessary for UAT. A separate user acceptance task should be created on ZenHub to track status of UAT. The internal product owner should close UAT task on ZenHub once the team receives test report from Professor. The development team will evaluate the test reports and will perform the necessary fixes. New Issues should be created on ZenHub to track these fixes. The team is expected to receive response from Professor within three days. Otherwise, the team would mark the UAT task complete on ZenHub.

Software Engineering Protocols

- Jasmine, Cucumber for BDD
- RSpec for TDD, Unit
- Travis for Continuous Integration(CI)

Test Plan

The purpose of the Test Plan is to gather all of the information necessary to plan and control the test effort for this project. The purpose of this document is to:

1. Identify the requirements that are to be tested.
2. Outline the testing approach that will be used.
3. Describe the workflow of the testing process that must be executed.

Test paradigm

Unit testing

The purpose of unit testing is to validate each unit of the software performs as designed. For this project, unit testing will be done in RSpec for models and controllers in Ruby with black box testing. Any unit test class should ensure that all non-trivial code paths are tested. Additionally, test class shouldn't rely too closely on the implementation. The developer will be responsible for writing down the tests for their model and controller changes. It will be necessary to have at least one test for each of your validations and at least one test for every method in your model. Coverage report should be generated after running the tests and should be at least 80%.

Integration testing

Integration testing aims to verify proper functioning when individual module works together. Therefore, any interaction that writes data or reads data from storage will require an integration test.

Additionally, integration testing ensures that the existing functionality will not break with the new changes. We have set up Travis CI on Github to do automated testing for new changes in the software. It will be activated whenever a pull request is created or merged into master. If it produces bug, an issue(T1-Defect) will be created on Github and will be assigned to the respective developer. If the reviewer finds out a possible bug in the pull request, he/she will have to notify the developer. The developers can work with reviewers through conversations and fixed the issues/concerns. The reviewer will have to create an issue(T1-Defect) on Github using bug template to document the bug.

Acceptance testing

The purpose of acceptance evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. Acceptance testing will be done in Cucumber. The product owner will decide "completeness" of the features based on 3 main criteria:

1. The functionalities of the features have been achieved
2. The specification for design have been achieved
3. The risk of software failure is reduced to a minimum

The team will have to address the concerns should it not meet his/her standards. The pull request will be merged into master once it is approved by the product owner. We will also use Cucumber - Capybara to automate the acceptance testing process for different use cases.

Functional testing

Functional testing makes sure that all the components of the application are working together properly. The developer will be responsible for writing down the test to make sure that that multiple components of the application such as databases, controllers and mailers are working in unison to handle the request and return the correct response. It will be done once the feature is completed.

Testing workflow

Bug Report and Change Requests

Whenever a bug is noticed while using the system, the development will submit a bug report on Github with label T1-Defect and level of severity of the bug. The template of the bug report is

Describe the bug

When pushed to production, file upload cannot load image url as well as link to the pdf viewer. It looks like the problem with Google drive permission and how the link generated from the website

To Reproduce

- Push the application to heroku
- Log in with valid credentials
- Go to profile page and try to upload picture
- The app will crash and heroku log will be show this error

```
2019-03-04T03:27:23.944756+00:00 app[web.1]: Caught error Forbidden: spp300-admin@spp300.iam.gserviceaccount.com does not have storage.objects.get access to store-faculty-id-card/profile/1/photo_id/661aa3fb65c8343e57dcab65be49ce0.png.
2019-03-04T03:27:23.945238+00:00 app[web.1]: Error - #<Google::Apis::ClientError: forbidden: spp300-admin@spp300.iam.gserviceaccount.com does not have storage.objects.get access to store-faculty-id-card/profile/1/photo_id/661aa3fb65c8343e57dcab65be49ce0.png, status_code: 403 header: #<HTTP::Message:Headers:0x00007fffb59ce0e00 @http_version="1.1", @body_size=0, @chunked=false, @request_method="GET", @request_uri="#>@AddressableURI:0x3fd1ace7583 URL:https://www.googleapis.com/storage/v1/b/store-faculty-id-card/o/profile%2F%2Fphoto_id%2F661aa3fb65c8343e57dcab65be49ce0.png>, @request_query=nil, @request_absolute_uri=nil, @status_code=403, @reason_phrase="Forbidden", @body_charset=nil, @body_date=nil, @body_encoding=<Encoding:UTF-8>, @is_request=false, @header_items=[{"X-GUploader-UploadID": "AEnbB2uQz24-XKbJH23_Z_0u8jCfIxrhk0CfPjGAvkfMuWe2v0Cn9-oMyLGx1yZzgGRjQXDMFshDfG022xWLFp0eSNW0BA"}, {"Vary", "Origin"}, {"Vary", "X-Origin"}, {"Content-Type", "application/json; charset=UTF-8"}, {"Date", "Mon, 04 Mar 2019 03:27:23 GMT"}, {"Expires", "Mon, 04 Mar 2019 03:27:23 GMT"}, {"Cache-Control", "private, max-age=0"}, {"Content-Length", "480"}, {"Server", "UploadServer"}, {"Alt-Svc", "quic=V:443"; ma=2592000; v="V44,43,39"}], @dumped=false> body: "{\"error\": {\"errors\": [\"\\n \\\"domain\\\" \\\"global\\\",\\n \\\"reason\\\" : \\\"forbidden\\\",\\n \\\"message\\\" : \\\"spp300-admin@spp300.iam.gserviceaccount.com does not have storage.objects.get access to store-faculty-id-card/profile/1/photo_id/661aa3fb65c8343e57dcab65be49ce0.png.\\n \\\"\\n \\\"code\\\" : 403,\\n \\\"message\\\" : \\\"spp300-admin@spp300.iam.gserviceaccount.com does not have storage.objects.get access to store-faculty-id-card/profile/1/photo_id/661aa3fb65c8343e57dcab65be49ce0.png.\\n \\\"\\n \\\"\\n \\\"}], @request_web_id: 2019-03-04T03:27:23.945241+00:00 app[web.1]: Completed 500 Internal Server Error in 499ms (ActiveRecord: 18.3ms)
2019-03-04T03:27:23.946906+00:00 app[web.1]: 2019-03-04T03:27:23.947718+00:00 app[web.1]: Google::Cloud::PermissionDeniedError (forbidden: spp300-admin@spp300.iam.gserviceaccount.com does not have storage.objects.get access to store-faculty-id-card/profile/1/photo_id/661aa3fb65c8343e57dcab65be49ce0.png.)
2019-03-04T03:27:23.947718+00:00 app[web.1]: app/controllers/profiles_controller.rb:81:in update'
2019-03-04T03:27:23.947720+00:00 app[web.1]:
2019-03-04T03:27:23.947721+00:00 app[web.1]:
```

Expected behavior

Image thumbnail should be able to generated

Once the issue is submitted, at least another person in the team will have to verify the bug. If it's impossible to verify it, the person creating the issue will either have to recreate the bug or close the issue. If the bug is verified, the team will decide whose code needs to change and fix it. The pull request to fix the bug must reference the issue and reviewed by the person who found the issue. Any pull request submitted needs to follow the following format:

This PR includes implementation for grading scale popover in application filtering page.

Issue Related

#248

Type of change

Please delete options that are not relevant.

- Bug fix (non-breaking change which fixes an issue)
- New feature (non-breaking change which adds functionality)
- Front end change

Code coverage

98.14% (Ruby)

Screenshot (if any change in front end)

Feature Acceptance Criteria and Process

The team is going to adopt the Red/Green/Refactor cycle for our testing approach. In the read phase, we will write a test on a behavior that we are about to implement representing how the code will be used by different clients. In the green phase, the programmer will write code to make the test pass. The last phase, refactor, we will remove any code duplication within our

program while keeping all tests green. We will also measure our C0 code coverage and aim to keep it above 90%.

Acceptance Criteria

- All the high priority test cases has been executed
- The source code has been unit tested
- There is no noticeable problem of the system found when introducing the new feature

Suspension Criteria

- Problem with testing infrastructure
- Significant change in project requirements

Resumption Criteria

- Problems cause the suspension have been resolved

Risk

- Delay in writing test case might require additional effort to deliver expected features by the end of the sprint
- Continuous changing requirements

Tools

We are using RSpec, a test driven development tool, for writing unit tests for model and controller. One of the greatest benefits of RSpec is its flexible mock-objects system and behavior-driven design. For acceptance testing, we are using Cucumber with integration with Capybara. For client testing, we are using Jasmine to test all of our Javascript component. For continuous integration, we are using Travis for automatic testing and deployment.

Useful materials

- [Travis CI installation for Rails](#)
- [Devise documentation for multiple devise user models](#)
- [File upload library in Rails](#)
- [Empathy in code review](#)
- [Data table for JQuery](#)
- [File upload with Shrine in Rails](#)
- [Add Jasmine testing to Rails Project](#)

Meeting Minutes

Sprint1

2019-01-23 In Person Meeting

Attended by Avanti, Linh, Julia and Harsha

- Two ideas came up under discussion
 - i. Parking application - facilitate users to reserve parking spot ahead of time.
 - ii. Platform for students to research and understand the process for applying to graduate schools based on their academic details.
- The first idea was dropped since we could not find enough features in order to scale the application. Also, we were not sure of the implementing GPS functionality.
- Discussed about the tools and technology to be used for project.

2019-01-25 In Person Meeting

Attended by Avanti, Linh, Julia and Harsha

- Discussed about the privacy aspect of the application users. The goal is to facilitate the users to restrict the visibility to their profiles if they want.
- Linh mentioned about the national database website which can help us extracting the data about graduate schools.
- Discussed about authenticating the faculty members
 - Harsha suggested to use faculty member's university page link to do this.
- Revised the scope of the project
 - Initially to start with two courses(CS and CSE) and then expanding it later.

2019-01-28 In Person Meeting

Attended by Avanti, Linh, Julia and Harsha

- Discussed about presentation for product vision.
- Discussed security implementation.

2019-01-30 Skype Meeting

Attended by Avanti, Linh, Julia and Harsha

- Created and discussed about user stories.
- Designed database tables with normalization.
- Assigned user stories to create GitHub issues.
- Talked about using Postgresql for development and testing environment.

2019-01-31 Skype Meeting

Attended by Avanti, Linh, and Harsha

- Selected and assigned estimates for following issues for Sprint 1 in ZenHub:
 - Create faculty table - 1
 - Create admin table and add admin user - 1
 - Student Account Sign Up - 2
 - Faculty account creation - 2
 - Account log out - 1
 - Account password reset - 2
 - Faculty log in - 1
 - Student log in - 1
 - Allow admin to approve faculty credential - 3
 - Collect data for university - 5
- Estimates were assigned based on prior experience of members.
- Decided that separate branch be used by members for development, then merged with pull request after approval by at least two members.
- Product owner for Sprint 1-Avanti and Scrum Master for Sprint 1-Linh
- Decided that the MVP for Sprint 1 is the application with successful login/logout and sign up for faculty and student.
- Decided to incorporate jasmine into testing, as Angular JS will be used. Cucumber and Rspec will be used for BDD and Unit testing.

2019-02-01 In Person Meeting

Attended by Avanti, Linh, Julia and Harsha

- Discussed about the story point estimation using ZenHub.
- Redefined our major user stories as EPIC and further defined simpler issues as subtasks on ZenHub.
- Updated backlog on ZenHub.
- Assigned tasks to the team for sprint1
 - Julia will work on student, admin and faculty UI.
 - Avanti and Harsha will work on controllers.
 - Linh will work on models.

2019-02-05 In Person Meeting

Attended by Avanti, Linh, Julia and Harsha

- The team worked on troubleshooting issues on angularjs front end + rails
- Linh created student model which will store student signup details.
- Avanti extracted data for some universities in .csv file and is still working on extracting data for universities.
- Julia is working on UI for home page and registration/login page.
- Harsha is working on faculty model/controller.

2019-02-07 In Person Meeting

Attended by Avanti, Linh and Harsha

- Worked on test plan presentation.
- Linh helped to debug environment issue on Harsha's machine.
- The team was able to get Angular and Devise work for the student login.

2019-02-06 In Person Meeting

Attended by Avanti, Linh, Julia and Harsha

- Talked about test plan presentation.
- Debug environment issue on Harsha's machine.

- Debug Devise issue with login.
- Avanti is extracting ranking and website for the university.
- Julia is working on Angular for UI for the login, sign up for students.
- Harsha is working on controller and model for faculty.
- Linh is working on Angular for UI for the login and signup for students.

2019-02-08 In Person and Hangout Meeting

Attended by Avanti, Linh, Julia and Harsha

- Team agreed to abandon AngularJS for the frontend due to following reasons:
 - Lack of documentation and integration support for AngularJS with rails.
 - Team's priority is to make steady progress to deliver a fully functional end product which should not be hindered by less supportive tools.
- Worked on Change Request Description for the Frontend Technology change.
- Team will work on haml/bootstrap and javascript for frontend.

2019-02-09 Hangout Meeting

Attended by Avanti, Linh and Harsha

- Harsha worked on ID_card upload functionality with Shrine gem.
- Linh worked on admin functionality with rails_admin gem.
- Avanti agreed to work on email confirmation setup for devise.

2019-02-11 In Person Meeting with Prof.Hans for Frontend Technology Change

Attended by Avanti, Linh, Harsha

- Explained the reasons for front-end technology change i.e. continue without AngularJS to Prof. Hans.
- Prof. Hans approved the change.
- Briefed professor about progress of project.
- Prof. Hans suggested to use Typescript in the project.
- Prof. Hans mentioned that he wishes to be a product owner for the project as he is in the graduate admission committee.

2019-02-12 In Person Meeting

Attended by Avanti, Linh and Harsha, Julia called in later to catch up

- Linh was able to figure out the file upload issue with Shrine gem.
- Team decided to use separate logins for student and faculty.
- Team created LowFi sketches for the nav-bar to separate logins.
- Harsha undertook to see how the file upload works in the production environment.
- Julia got caught up with the current progress and working on the front end of homepage and integrating CSS into our project.

2019-02-13 In Person Meeting

Attended by Avanti, Linh, Harsha and Julia

- The team discussed switching the Github repo to become public since Zenhub trial is expiring.
- Avanti reviews current status of tasks on Zenhub.
- Julia is working on improving the front end for the homepage and fixing her PR.
- Harsha created a Google Cloud account for storage on production and setting up production environment in Heroku.
- Linh created a bug report on the issue of cross model login and merged the PR for outlining cucumber.

2019-02-15 Hangout Meeting

Attended by Avanti, Linh, Harsha and Julia

- The team helped Harsha with uploading files to Google Cloud.
- The team helped Julia with fixing merge conflicts and removing unnecessary files from her PR.

2019-02-17 Sprint 1 Retrospective

Attended by Avanti, Linh, Harsha

- The team discusses accomplishments of Sprint 1
 - We were able to finish the MVP for Sprint 1
 - Deployed our app to Heroku
 - Implemented file upload to Google Cloud Storage
 - Debug sessions and frequent check in

- The team discusses failures we met during Sprint 1
 - Testing protocol hasn't been followed consistently. Some PR merged into master didn't include proper testing
 - Pull request and commit message doesn't contain informative description
 - Zenhub planning wasn't well-organized, didn't keep track progress on task closely
- Things we can do to improve
 - Create pull request template
 - Any pull request need to include proper testing and report code coverage

Sprint 2

2019-02-18 Hangout Meeting

Attended by Avanti, Linh, Harsha and Julia

- The team has decided to work on following features for Sprint 2
 - Creation of student profile
 - Look up university information
- Designed lofi sketches for the above features.
- Linh and Julia will work on creation of profiles.
- Avanti and Harsha will work on search for university information.
- Wiki page is updated for Sprint 2 overview.
- Decided Product Owner for Sprint 2-Harsha and Scrum Master for Sprint 2-Avanti.
- Story points are estimated and epics/issues are created on Zenhub.

2019-02-20 In Person Meeting

Attended by Avanti, Linh and Harsha

- Zenhub was reviewed for the current status of task assigned team members.
- Linh worked on creation of profile and application model.
- Avanti worked on university model and seeding information into university table.
- Harsha is working on using jtable-rails for searching and filtering universities based on rank and university type.

2019-02-22 Hangout Meeting

Attended by Avanti, Linh, Harsha and Julia

- The team helped Avanti to align university fields on show UI for displaying university information.
- The team helped Harsha to fix display issues with datatable gem.
- Linh is working on show UI of creation of student profile.
- Avanti worked on adding university records in university table. The count of the universities has gone up to rank 40.

2019-02-25 In Person Meeting

Attended by Avanti, Linh, Harsha and Julia

- The team had a meeting with Prof. Hans Johnson to discuss about the faculty usage of the application. Prof. Hans provided lofi sketch for the faculty evaluation of applicants and different academic fields required for evaluation.
- The team has decided to work on faculty component of application from sprint3.
- The team should show functionality of application(faculty) to graduate committee sometime at end of April.
- Linh is working on the functionality of allowing students to edit their profile.
- Julia is working on creating user interface for adding universities on students profile.
- Harsha is working on BDD for searching and filtering universities.
- Avanti is working on BDD for showing university details of universities.
- Zenhub was reviewed for the current status of tasks assigned to team members.

2019-02-27 Hangout Meeting

Attended by Avanti, Linh and Harsha, Julia was having technical difficulty joining Hangout

- Harsha updated that he is working on the issues BDD for university search and filter, and he is struggling with BDD for university search as Capybara does not recognize javascript modified DOM elements. However he agreed to integrate another tool such as Jasmine to overcome that and make a PR for them by the end of Thursday.
- Linh updated that she is working on the issue Upload student photo and other attachments, and that she would need Harsha's support as he is the one created the cloud storage account.
- Avanti updated that she is working on the issue BDD for showing university details on UI and got assurance from Harsha that universities index view is functioning properly.
- Avanti also updated that she was able to extract and write almost 97 university records to the database and that we should be able to close the issue Import details of university into database by the end of Sprint 2, with several more new records in the next few days.

2019-03-01 Hangout Meeting

Attended by Avanti, Linh, Julia and Harsha

- Avanti updated that she is struggling to fix the double click issue related to jquery datepicker. The calendar on date column was visible only after second click on calendar icon. The team helped Avanti to troubleshoot this issue. Finally, Julia has agreed to look more into it.
- Linh is working on creating a download link for SOP, resume and other attachments on student profile.
- Harsha is working on setting up jasmine tests for university search.
- The team has decided on meet on hangout on Sunday @11 am to discuss about Sprint 2 retrospective.

2019-03-03 Hangout Meeting

Attended by Avanti, Linh and Harsha

- Harsha was having problem with capybara couldn't recognize javascript element. The team together debugged the problem and found the solution is to include the data table javascript library into the application.html.erb file so that the capybara can recognize the element. Linh opened the PR for that issues and merged to master.
- The team also have a discussion on sprint 2 retrospective. Some of our accomplishments include
 - Finished MVP for creating/editing student profile, collect university data and was able to populate it in the UI. Also, we added functionality to search and filter university by ranking and name.
 - Able to follow testing protocol by having code coverage ~90% for all of the PRs on sprint 2.
 - Updated issues on Zenhub timely so burndown chart looks a lot better than sprint 1.
- However, some of our failures are
 - Overestimate amount of work we can accomplish. We have 7/55 unfinished story points.
 - Some of our work was done in pair so some PR contains commits of more than one person.
 - It was more difficult to schedule meeting this sprint due to people's schedule.
 - Some of our issues on Zenhub does not include good description.
- Therefore, for next sprint, we want to
 - Set better velocity points for the sprint. With 38 completed from sprint 1 and 48, we decide 45 - 50 is a realistic goal for the next sprint.
 - Set more time to write description for github issues.
 - Document pair work on PR description if the PR contains commits from more than 1 person.
- We also did a planning for sprint 3. Linh is the product owner and Harsha is scrum master. The goal for implementation for the next sprint will be

- Finish work from Sprint 2, which will implement potential school list for student.
- Create grading schema for college, undergraduate school model for student profile.
- Populate students information for faculties evaluation.
- We decided to meet tomorrow with Prof to discuss on our backlog and protocol for story acceptance at 10:30am.

Sprint 3

2019-03-04 In person Meeting

Attended by Avanti, Linh, Harsha and Julia

- The team met with Prof to ask questions about user acceptance protocol. Based on his instructions, we need to come up with a criteria for product owner to approve features.
- The team prepared for sprint 2 retrospective presentation.
- We delegated task for sprint 3
 - Avanti will be working on potential school list.
 - Harsha will fix his environment issue and work on showing application for faculty views.
 - Linh will be working on database for university.
 - Julia will work with data table for university.
- Harsha and Linh will meet tomorrow to work on the bug for file upload in production.

2019-03-06 Hangout Meeting

Attended by Avanti, Linh, Harsha and Julia

- Avanti is working on potential school list.
- Harsha fixed his environment issue and work on showing application for faculty views.
- Linh and Harsha fixed issue with file upload in production and is working on add university field to faculty table.
- Julia is working with data table for university.

2019-03-08 Hangout Meeting

Attended by Avanti, Linh, Harsha and Julia

- Avanti made progress with university list and will create a PR for it today. She's having problem with showing flash message on reload page.
- Harsha is working on controller method to show applications for faculty's view.
- Julia is working with data table for university and will collaborate with Harsha on this.
- Linh is working on adding undergraduate model.

2019-03-11 Hangout Meeting

Attended by Avanti, Linh, Harsha and Julia

- Linh was able to help Harsha to help fixing the Cucumber tests that were failing.
- Avanti requested help from Harsha to add a pop up window to edit schools in the student's profile.
- Team agreed to work during during the spring break.
- Team agreed to have the next meeting on Wednesday.

2019-03-13 Hangout Meeting

Attended by Avanti, Linh, Harsha and Julia

- Harsha was having trouble with the jquery datepicker and Avanti suggested to use html input type=date instead.
- Team discussed that in the edit school, dates have to be validated to prevent applied date being empty.
- Harsha agreed he will work on edit school with html date fields and add javascripts.
- Avanti was struggling with javascript testing as DOM is modified by javascript. Harsha was experiencing the same. Only Linh was able to get the tests pass.
- Linh informed that she won't be able to attend next meeting on Friday.

2019-03-15 Hangout Meeting

Attended by Avanti and Harsha

- Avanti was struggling with the BDD of deletion of schools as javascript is not getting executed on delete icon click. Harsha has agreed to take a look into it.
- Harsha is working on adding additional filters for viewing students profiles by faculty members.
- Avanti has agreed to write down rspec for creating and showing students profiles method.

- Linh could not attend the meeting as she is in transit today.

2019-03-18 Hangout Meeting

Attended by Avanti, Linh and Harsha

- Harsha was having trouble with filtering the applications of students by faculty members based on undergrad universities. The team together debugged the problem and created an association between applications and profiles to filter out the students with the input undergrad universities.
- Currently, admin can see only the metadata of faculty i-card. Hence, Linh is working on displaying faculty i-card upload to admin so that admin can view it.
- Avanti is working on BDD of deletion and edit application status of schools.

2019-03-20 Hangout Meeting

Attended by Avanti, Linh and Harsha

- Avanti was helping Harsha with writing RSpec for filtering applications.
- The team discussed about potential change in database structure for undergraduate university. Students should be able to add their major and gpa depending on their undergrad institution.
- Avanti found a bug with filtering application. Harsha will look into it.
- Linh opened a PR to parse back end error for adding undergrad university. She will open another PR to fix the vulnerability alert issue.
- Avanti is working on Jasmine for my applications page.

2019-03-22 Hangout Meeting

Attended by Avanti, Linh and Harsha

- Harsha was having trouble with the BDD of filtering of applications based on research interest by faculty members. The team together debugged the problem and resolved it.
- Avanti is struggling to mock school modal in jasmine while editing school applications. Linh has agreed to look into it.
- Harsha informed that he will create a PR to filter the applications by faculty members tomorrow.

- The Team has agreed to meet on Sunday @4 pm.

2019-03-24 Hangout Meeting

Attended by Avanti, Linh and Harsha

- The team did sprint 3 retrospective and planning for sprint 4
- What went right:
 - Collaboration between member are documented.
 - Issue description on Zenhub are more clear and well-organized.
- What went wrong:
 - Javascript testing and Capybara didn't work well together.
 - We couldn't capture JS test code coverage.
- The Team has agreed to meet on Monday @11:30 am.

Sprint 4

2019-03-25 Hangout Meeting

Attended by Avanti, Linh and Harsha

- The team discussed about setting up mock profiles on production heroku application for user acceptance testing.
- The team has agreed to create five mock student profiles and one faculty account to make sure that external product owner can filter profiles based on different academic criteria.

2019-03-27 Hangout Meeting

Attended by Avanti, Linh, Julia and Harsha

- Linh confirmed that the filtering by faculty is now working fine.
- Team decided to keep only the issue 'creation of the evaluation table' in the sprint 4 and move 'evaluation form creation' to sprint 5.
- Linh asked if Julia could modify the edit profile and show profile pages to look better in terms of UI/UX and Julia agreed to look into that.
- Harsha agreed to add necessary javascripts for Add Undergrad School form to avoid double selections in drop downs.
- Team discussed about the contents of the Friday Mid semester presentation.
- Avanti agreed to initiate the presentation for Friday, while others will add content to that.
- Harsha agreed to push the master branch to heroku and create faculty account for Prof. to perform acceptance testing for Application filtering.

2019-04-03 In Person Meeting

Attended by Avanti, Linh, Julia and Harsha

- The team discussed about the current status of their work.
- Harsha has completed UI as well as back-end changes for adding term and year filtering for faculty members and is working on the BDD for the same.
- Linh has completed the feature to allow faculty members to view student's profile and is working on the BDD for the same.
- Julia has completed UI changes for adding header fields while adding schools by the students.
- Julia will be working on creating evaluation form for faculty members.

- Avanti is working on adding jasmine tests for myapplications page.
- The team has decided to meet on Friday(5th April) @11:30 pm.

2019-04-05 Hangout Meeting

Attended by Avanti, Linh, Julia and Harsha

- The team discussed about the current status of their work.
- Harsha opened and merged PR to add UI as well as back-end changes for adding term and year filtering for faculty members and is working on the BDD for the same. He will start working on adding Google Maps API to the project.
- Linh has completed adding BDD tests for faculties viewing students application and profiles.
- Julia is working on UI for submitting faculty evaluation for application.
- Avanti is working on adding jasmine tests for myapplications page. She also opened a PR to fix an issue with adding application by term and year
- The team has decided to meet on Sunday(7th April) for sprint 4 retrospective

2019-04-07 Sprint 4 Retrospective

Attended by Avanti, Linh and Harsha

- The team discussed on sprint 4. We were able to finish most of the points planned in the beginning of the sprint. We were able to find teaspoon to add the code coverage for JS. We also created the UAT and sent it to Professor. However, we still had problems with Jasmine tests mocking.
- We also planned for the sprint 5 task. For this sprint, we will focus on creating the discussion forum, the prediction model for students' admission chance and faculty evaluation form. Number of velocity planned for sprint 5 is 45.
- We will meet on Monday 4/8 for Scrum meeting.

Sprint 5

2019-04-08 Hangout Meeting

Attended by Avanti, Linh and Harsha

- Avanti informed that she has started working on viewing evaluations by faculty.
- Linh informed that she has started working creating students discussion forum.
- Harsha informed that he has started working filtering of applications based on grading scales.
- The team will have the next meeting on Wednesday 4/9.

2019-04-10 Hangout Meeting

Attended by Avanti, Linh and Harsha

- Avanti has finished feature - view evaluations by faculty members and has started working on bdd for the same.
- Linh has finished working creating discussion forum and opened a PR for it.
- Harsha informed that he will split percentage fields into two fields 'low_percentage' and 'high_percentage' in grading scales table in order to use these fields whenever faculty performs filtering of applications based on grading scales percentage.

2019-04-12 Hangout Meeting

Attended by Avanti, Linh, Julia and Harsha

- Avanti is struggling with aligning back button on view other evaluations page. Harsha has agreed to look into it.
- Linh has completed BDD for posting and answering to queries on forum.
- Harsha is working on applications filtering by grading scale and will be able to open a PR for the same by tomorrow.
- Julia is working on the UI for suggesting universities as dream, target and safe for students.
- The team has decided to add grading scales information popover on student profile whenever a faculty views a student profile. Harsha will be incorporating this new change in his PR.

2019-04-15 Hangout Meeting

Attended by Avanti, Linh, Julia and Harsha

- Avanti is working on Jasmine for my application page.
- Linh is working on adding dynamic research interest in edit profile page.
- Harsha is working on rspec for faculty filtering application by grading scale.
- Julia is working on the UI for suggesting universities as dream, target and safe for students and will open a PR by tomorrow.
- The team worked on the testing presentation for Wednesday. We decided to meet 10:30 am on Wednesday for next meeting

2019-04-19 Hangout Meeting

Attended by Avanti, Linh and Harsha

- Harsha has finished working on faculty filtering application by grading scale and will be working on BDD for the same.
- Linh has finished working on adding dynamic research interest on edit profile page and has opened a PR for the same. Next, she will be working on BDD for the same.
- Avanti informed that she will open a PR for jasmine tests mostly by tomorrow.
- Avanti has opened a PR for adding some more functional tests for models and controllers.
- Julia has finished working on the UI for suggesting universities as dream, target and safe for students.
- The team has decided to meet on Sunday 04/19.

2019-04-21 Hangout Meeting

Attended by Avanti, Linh and Harsha

- The team have a discussion on sprint 5 retrospective. Some of our accomplishments include
 - Successfully implemented MVP for this sprint
 - Creation of discussion forum for students
 - Application filtering based on different grading scales
- However, our failure is
 - We faced issues in mocking some of the function calls while writing down jasmine tests for javascript. Hence, our jasmine coverage is only 75%.

- The team also did a planning for sprint 6. Avanti is the product owner and Linh is the scrum master. The goal for implementation for the next sprint will be
 - Implementation of prediction model
 - Adding dynamic research interest for students
 - Adding grading scale popover to students profile for faculty members to view it
 - Poster presentation preparation and project documentation
- Linh will update Sprint 5 discussion/failures on wiki page.
- Harsha will pushed the sprint 5 changes to heroku and will update sprint 6 overview on wiki page.
- Avanti will create issues for sprint 6 on Zenhub.
- The team has decided to meet on Monday 04/22.

Sprint 6

2019-04-22 Hangout Meeting

Attended by Avanti, Linh and Harsha

- Harsha is going to work on the popover about grading scale on faculty page.
- Linh is going to work on fixing the PR for adding dynamic research interest.
- Avanti is working on the UAT document to send it to the Professor to perform UAT on faculty related features.
- The team has decided to meet on Wednesday 04/24.

2019-04-24 Hangout Meeting

Attended by Avanti, Linh, Julia and Harsha

- Harsha is working on the pop over of grading scale. Harsha pushed new version of the app to Heroku and found a bug on faculty evaluation page. An issue has been created for this bug and Linh is going to work on it today.
- Avanti has finished UAT and is going to work on the prediction model. Harsha has suggested a formula to calculate the prediction based on GRE scores.
- Julia is going work on the poster presentation.
- Linh informed that she won't be able to attend next meeting as she has wisdom tooth surgery on Friday.
- The team has decided to meet on Friday 04/26.

2019-04-26 Hangout Meeting

Attended by Avanti, Julia and Harsha

- Avanti mentioned that she has made a Pull Request for Chance Prediction and that she has fine-tuned the prediction since yesterday.
- Team appreciated the Gremester logo and poster template shared by Julia.
- Harsha confirmed that he has finished the grading scale popover and will write a jasmine test for that.
- Team clarified what is expected of an API call for chance prediction.
- Team agreed to work on abstract and poster on the google drive.

- Team decided to meet on Sunday 4/28 evening.

2019-04-28 Hangout Meeting

Attended by Avanti, Linh and Harsha

- Harsha informed that he will create a PR for grading scale popover later in the evening.
- Team discussed about the project report and worked on abstract and project overview of the project.
- Linh suggested to add database schema into project report which will be generated using diagram options of rails framework.
- Team decided to close the UAT task assigned to Prof.Hans in Zenhub.
- Team has decided to meet on Tuesday 04/30.

2019-04-30 In Person Meeting

Attended by Avanti, Linh, Julia and Harsha

- Harsha created and closed the PR for grading scale popover later in the evening.
- Harsha found a bug in the research interest filtering and will create a PR to fix the bug.
- Julia will make few layout changes in the poster and will share the final version with the team to review before printing it.
- The Team worked on poster presentation today and will meet tomorrow to finalize the project report.

Sprint 1- 6 Overview

Sprint 1

G003 - Gremester Sprint 1 Retrospective

Avanti Deshmukh - Product Owner
Linh Pham - Scrum Master
Julia Chaloupka
Harsha Pitawela

What went right?

- Early Identification of roadblock to our progress
 - Decided to discontinue using Angular as our main front end framework
- Debugging session with team members for good team dynamics
- Code coverage - 91.11 %
 - RSpec - 18 tests
 - Cucumber - 51 scenarios

What went wrong?

- Testing protocols
 - Some of merged pull requests doesn't include proper testing
- Project management through Zenhub
 - Initial backlog wasn't well-organized
 - Progress on tasks wasn't updated in time
- Naming conventions for pull request and commit messages
- Front end technology change issue



Burndown Chart Sprint 1

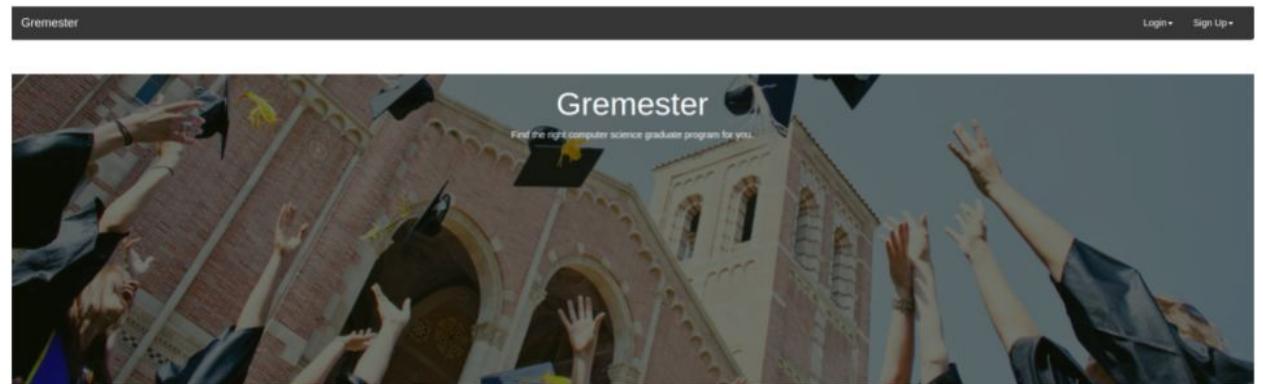


Velocity Points

- 42 initial points estimate
- 38 points adjusted after changing technology stack
- 33 points completed by the end of sprint 1



Accomplishments



Sign Up

Username

First name

Last name

Email

Password (8 characters minimum)

Password confirmation

[Log in](#)

[Didn't receive confirmation instructions?](#)



Gremester Admin

NAVIGATION: Admins, Faculties, Students

Site Administration

Dashboard

Dashboard

Model name	Last created	Records
Admins	1 minute ago	<input type="button" value="View"/>
Faculties	<input type="button" value="View"/>	<input type="button" value="Edit"/>
Students	1 day ago	<input type="button" value="View"/>

Gremester Admin

NAVIGATION: Admins, Faculty, Students

Details for Faculty 'Faculty #1'

Dashboard - Faculties - Faculty #1

Approve faculty Show Delete

Basic info

First name: phamlinhtu@gmmail.edu

Middle name: Linh

Last name: Pham

What we need to change for next sprint?

1. Enforce testing and code coverage report for every pull request
2. Create pull request template to help code reviewers understand the context better
3. Review Zenhub at the beginning of daily stand up
4. Engage more pair programming

Sprint 2

G003 - Gremester Sprint 2 Retrospective

Harsha Pitawela- Product Owner
Avanti Deshmukh - Scrum Master
Julia Chaloupka
Linh Pham

What went right?

- Finished implementation for creating and editing student profiles
- Added search and filtering functions for universities
- Added pull request templates to provide more context of the code change and monitor test coverage
- Timely update for Zenhub issues, which helps maintain more steady burndown rate than Sprint 1
- Met with our external product owner



What went wrong?

- Overestimation of velocity points
 - Completed 47/54 points
- Missing GitHub Issues descriptions
- Interleaved commit history from more than one team member for some pull requests

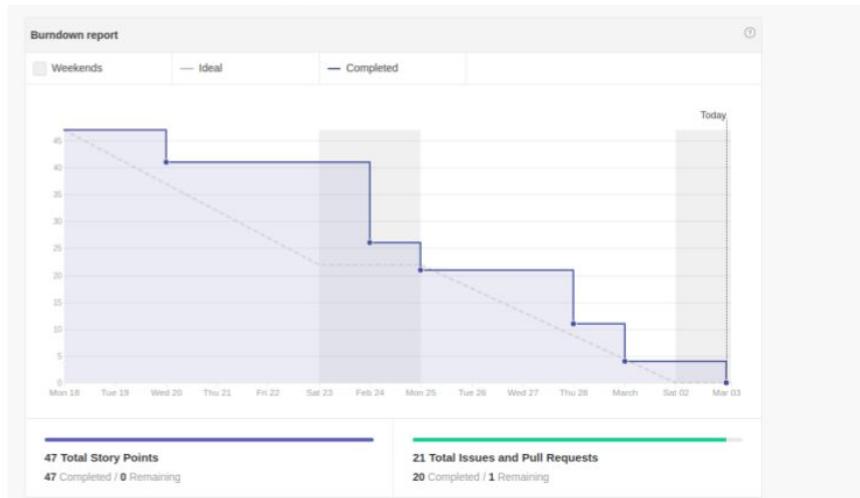


Testing

- Code Coverage: 95.53%
- Number of tests:
 - 43 new RSpec tests
 - 32 new Cucumber scenarios
 - 4 new Jasmine tests



Burndown Chart Sprint 2



What we need to change for next sprint?

- Documentation of GitHub issues properly
- Documentation of commits of multiple authors in pull requests

Sprint 3

G003 - Gremester Sprint 3 Retrospective

Harsha Pitawela- Product Owner
Avanti Deshmukh - Scrum Master
Julia Chaloupka
Linh Pham

What went right?

- Finished our MVP (52/ 56 velocity points)
- Enforced protocol to document co-authors on GitHub commits
- Allocated more time to plan Zenhub issues

What went wrong?

- Having difficulties with Jasmine testing



What we need to change for next sprint?

- Looks into ways to measure code coverage for Javascript code portion of the project

Mid-term Review

G003 - Gremester Mid-Semester Project Milestone

Avanti Deshmukh
Linh Pham
Harsha Pitawela
Julia Chaloupka

Sprint1 Milestone

- Implementation of user authentication protocols for faculties, students and admin
- Implementation file upload from production application to google cloud storage
- Integrated Travis into our GitHub Project
- Decided to stop using Angular for front end technology

Sprint2 Milestone

- Finished implementation for creating and editing student profiles
- Added search and filtering functions for graduate schools and programs
- Added pull request templates to provide more context of the code change and monitor test coverage
- Timely update for Zenhub issues, which helps maintain more steady burndown rate than Sprint 1
- Met with our external product owner



Sprint3 Milestone

- Implementation of applications filtering by faculty members based on a range of criteria
- Allow students to edit and delete universities through my applications
- Allow students to add their education to their profile dynamically



Testing

- Code Coverage: 93.22%
- Number of tests:
 - 105 RSpec tests
 - 104 Cucumber scenarios
 - 9 Jasmine tests



Burndown Chart Sprint 1



Burndown Chart Sprint 2



Burndown Chart Sprint 3



Cumulative Burndown



Lessons Learnt

- Importance of Feasibility study
- Value of Pair Programming
- Importance of Regular meetings
- Importance of frequent and small commits



Future Sprint Tasks

- Faculty evaluation for students' application
- Create discussion page for students posting questions
- Implement prediction model for students' chance of admission



Sprint 4

G003 - Gremester Sprint 4 Retrospective

Harsha Pitawela- Product Owner
Avanti Deshmukh - Scrum Master
Julia Chaloupka
Linh Pham

What went right?

- Successfully installed Teaspoon to measure Javascript code coverage
- Created user acceptance testing document and sent it to our external product owner

What went wrong?

- Still having problems mocking ajax requests using Jasmine



Burndown Chart Sprint 4



What we need to change for next sprint?

- Increase Jasmine coverage for Javascript

Sprint 5

G003 - Gremester Sprint 5 Retrospective

Avanti Deshmukh - Product Owner
Harsha Pitawela - Scrum Master
Julia Chaloupka
Linh Pham

What went right?

- Successfully completed MVP including adding discussion forums, faculty evaluation form and filtering by grading scale for faculties.
- Completed 41/45 velocity points

What went wrong?

- Test coverage for Javascript portion is still very low (75%)



Burndown Chart Sprint 5

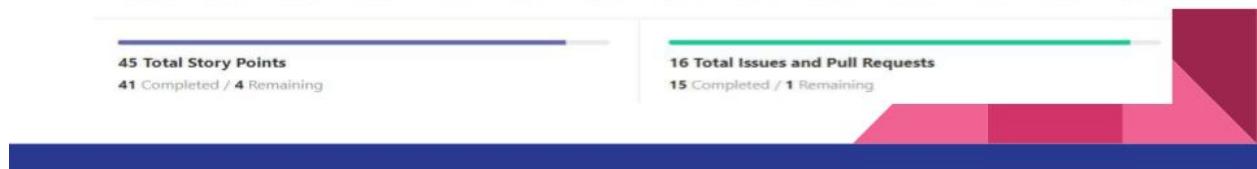


45 Total Story Points

41 Completed / 4 Remaining

16 Total Issues and Pull Requests

15 Completed / 1 Remaining



What we need to change for next sprint?

- Increase Jasmine coverage for Javascript

Progress Review and Testing Overview

G003 - Gremester Progress Review and Testing Overview

Avanti Deshmukh
Linh Pham
Harsha Pitawela
Julia Chaloupka

Features completed since mid-term review

- Finished implementation of discussion forum that allows student to post and answers questions
- Finished implementation of faculty evaluating students' application
- Finished implementation of faculty to view their own evaluations
- Finished implementation of faculty to view evaluations of other faculty members for a student profile
- Added Teaspoon for jasmine code coverage

Protocols

- Any PR needs at least 2 approving reviews to be merged
- Reviews are submitted within 24 hours since the PR is opened
- PR must include description of changes made, issues related, code coverage and screenshots if available
- Used bug report template whenever a bug is discovered by anyone in the team.



Future Sprint Tasks

- Continue working on Jasmine tests
- Get user acceptance tests
- Implementation of prediction model for students' admission chance
- Have non-developer's test UI for bugs and issues
- Looked into ways for performance/stress testing for the application



Gremester: A Graduate School Application Interface

THE
UNIVERSITY
OF IOWA

Linh Pham, Julia Chaloupka, Harsha Pitawela, Avanti Deshmukh
Software Engineering Project (ECE:5830) Spring 2019

Appendix

Poster Presentation

Introduction

Gremester is a web-based collaborative platform for prospective graduate students and university faculty members during the admission process. Gremester was created to streamline the workflow of the admission process. The app helps students maintain all their academic records used to apply to graduate school in one portal. Gremester is also useful for faculty members who do not have a web-based system for the application review process. In one platform, faculties can evaluate student profiles and record their evaluations.

User Interface:

Student Applications View

Faculty Application Evaluation

University Admission Predictor

Architecture

Architecture

```

graph TD
    Browser[Browser] <--> WebServer[Web Server]
    WebServer --> Routing[Routing]
    Routing --> View[View]
    View --> Controller[Controller]
    Controller --> Model[Model]
    Model --> Database[Database]
    Database --> Cloud[Cloud]
    
```

Features

- Create student profile with personal info
- Look up details about top 100 US graduate schools
- Add potential graduate universities to students profile
- Prediction of students' chance of admission to any university
- Discussion forum for students
- Faculty evaluation of student applications by faculty
- View undergraduate university location using Google Maps
- Administrator interface for website maintenance

Technology & Tools

Technology & Tools

- Rails
- Cucumber
- Jasmine
- Google Cloud
- SimpleCov
- Heroku

▲ test coverage: 100%

Methodologies

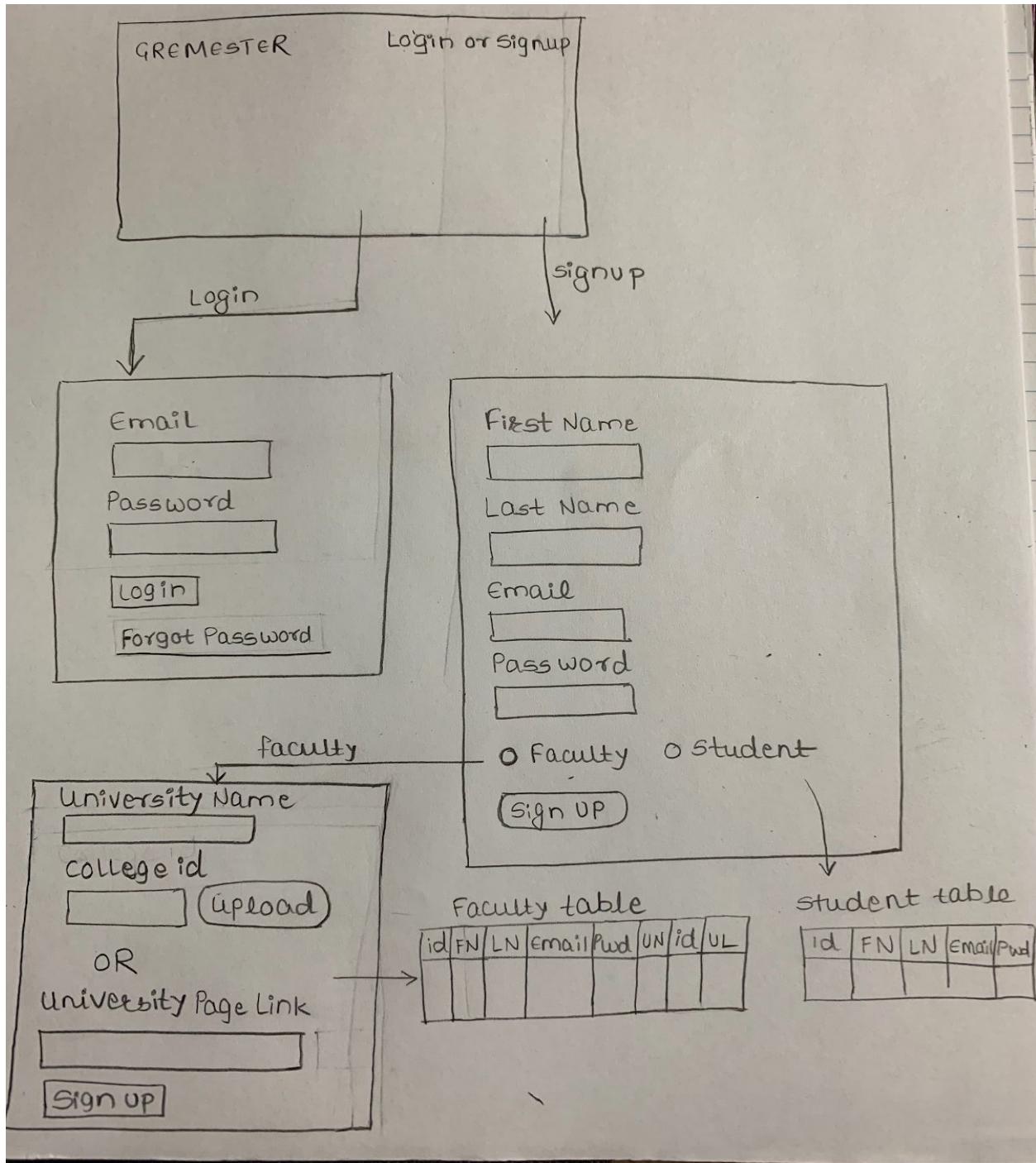
- Agile/Scrum Practices
- Behavior-Driven Development
- Test-Driven Development

Acknowledgement

We would like to thank our external product owner, Prof. Hans Johnson, and Niharika Anirudha (GCP student representative) for all the help throughout the semester.

Lofi Sketches

Gremester Signup/Login



Student Profile

My Profile

GRE MESTER	US STATE	USERNAME
<input type="text"/> Photo	<input type="text"/> FIRST LAST NAME	<input type="text"/>
	<input type="text"/> USERNAME	
INTERESTED MAJOR TERM		
PROFILE		<input type="button" value="EDIT"/>
GRE	TOEFL	CGPA
<input type="text"/>	<input type="text"/>	<input type="text"/>
INTERESTED MAJOR	TERM	YEAR
<input type="text"/>	<input type="text"/>	<input type="text"/>
COMPANY NAME	WORK EXP	
<input type="text"/>	(YEAR)	(MONTHS)
UNDERGRAD COLLEGE		
<input type="text"/>		
RESUME +		
SOP +		
Additional attachment		

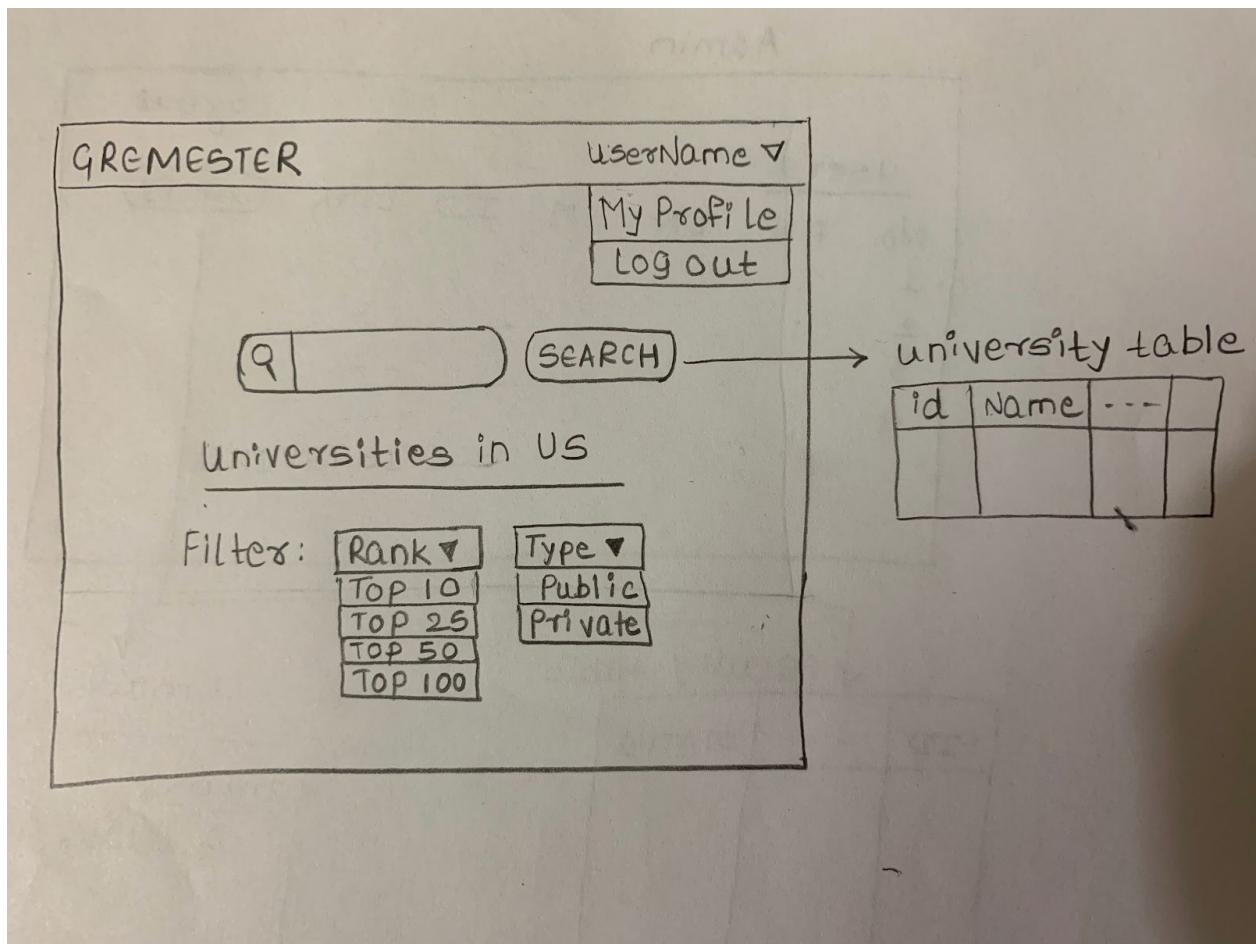
Universities applied

University name	status
<input type="text"/>	<input type="text"/>

↓ profile table

id			-
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Search Universities



Faculty to view student profile

Faculty to view student Profile

Gremester Hi, username ▾
Logout

Show entries Search:

Name	UndergCollege	ResearchArea	GPA	GRE(Q)	GRE(V)	MS/PHD

1 | 2 | 3 | Next

Profile Database
Undergrad universities database
Application Database

My Applications page

Schools of Interest page

Greemerter Hi, username

Schools of Interest

University Name	Applied Date	Admit Date	Reject Date
Boston University	03/07/2019		

(Add school)

↓

university	status	Date	Delete	Save
Dropdown for universities	Dropdown for options			

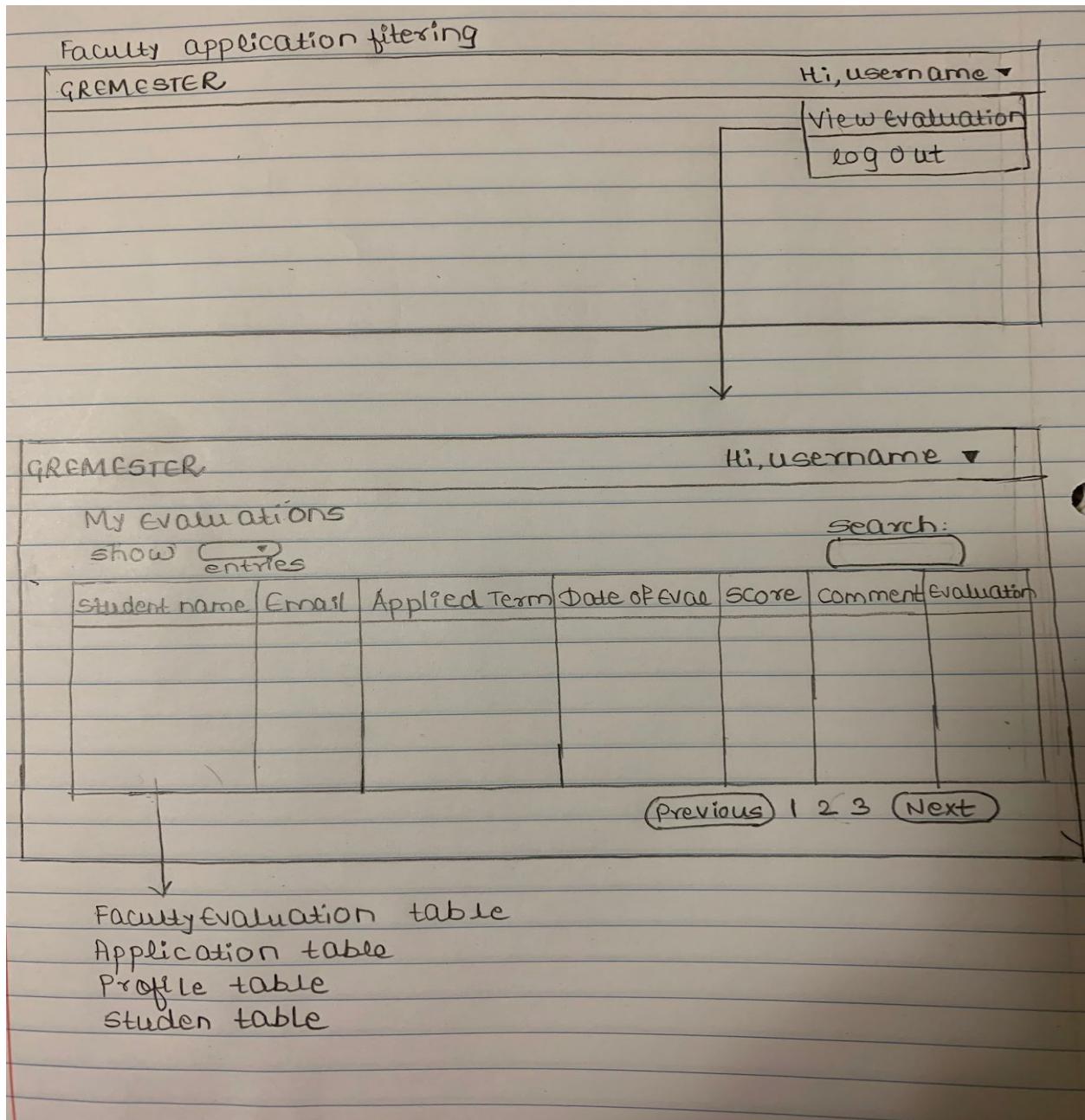
Application database

→ Edit school

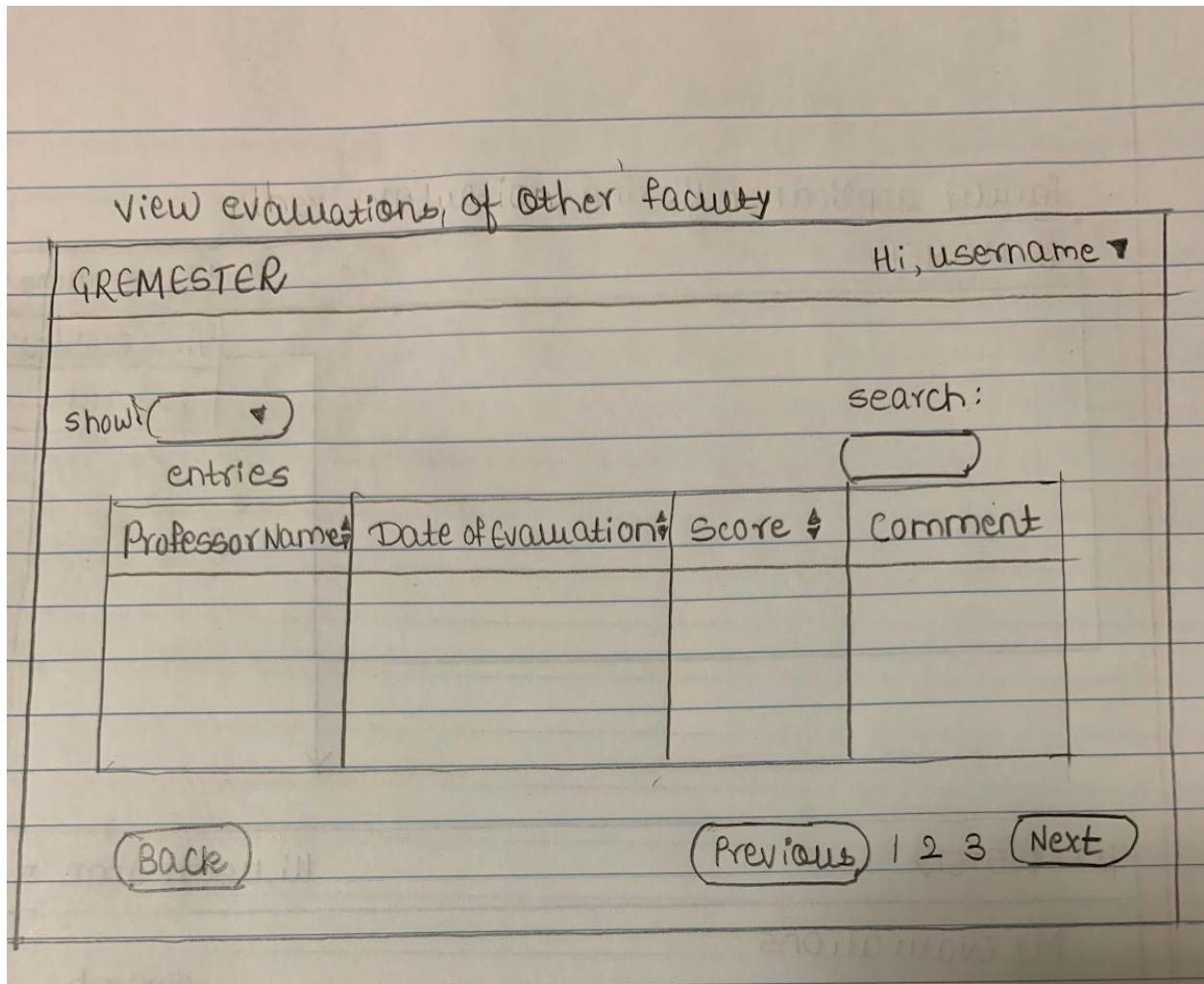
university Name	
Applied Date	
Accepted Date	
Rejected Date	
(Cancel) (Save)	

Application database

Faculty application filtering and view evaluations



Faculty to view other faculty evaluations



Admission Predictor page

The image shows a mobile application interface for 'GREMASTER'. At the top, it says 'GREMASTER' and 'Hi, Lily ▾'. On the left, there's a vertical menu with 'Edit Profile', 'My Applications', and 'Admission Prediction' (which has an arrow pointing to it). In the center, there's a table with columns for 'University Name', 'Admission Category', and 'Admission chance'. Below the table are three buttons: 'University ▾', 'Calculate', and a large downward arrow. The table data is as follows:

University Name	Admission Category	Admission chance
University of Iowa	Safe	84 %
Iowa State	Target	71 %
Harvard	Dream	47 %

Below the table is the text 'Profiles Controller' with a downward arrow.

Faculty to evaluate students

Faculty Evaluation

GREMASTER Hi, James ▼

Application Review

Applicant score
score ▼

EE background
EE background ▼

Comments

Submit

↓

Faculty evaluation database

Undergraduate details

GREMASTER Hi, Lily ▼

My Profile	Edit Profile
My Applications	Undergraduate: Add education, School
Admission Predictions	<input type="button" value="Country ▼"/> <input type="button" value="University Name ▼"/> <input type="button" value="Major ▼"/> <input type="button" value="GPA ▼"/> <input type="button" value="Grading scale ▼"/> <input type="button" value="Start year ▼"/> <input type="button" value="Exp. Grad Year ▼"/>

↓

Profiles-undergrad-university
database.



GREMESTER_MAINRELEASE 1.0

User Acceptance Test Document

Title:

GREMESTER_MAINRELEASE 1.0

Objective:

The purpose of Gremester is to provide a platform for potential graduate students to find suitable computer science graduate programs and members of admission committees to find suitable candidates for their graduate programs. The end-users of the system are potential graduate students and members of admission committees of computer science graduate programs.

Scope:

Scenarios related to faculty login and application filtering by faculty.

Introduction:

Requirements were gathered from users representing admission committees. Application filtering functionality has been implemented as per those requirements. In the UAT (User Acceptance Testing) phase the scenarios in this functionality will be tested and accepted/rejected by actual users. Test faculty account has been created for UAT.

Features to be Tested:

1. Email verification for faculty account
2. Faculty login
3. Application filtering by faculty on a range of criteria

Features not to be Tested:

1. Application evaluation by faculty

Approach:

Once all tests are performed, the development team will evaluate the test report and do bug fixes if necessary.

Test Environment Details:

Testing can be done by logging into <http://gremester.herokuapp.com>

Test faculty account credentials:

E-mail: hans-johnson@uiowa.edu

Password: 12345678

Tests:

#	Acceptance Testing Action	Pass/Fail	Date	Initials
1	Faculty cannot login before e-mail verification. Steps: Click Login > Faculty Login with above credentials			
2	Faculty can login after e-mail verification. Steps: Log into e-mail. Locate e-mail from gremestersep2019@gmail.com (check the junk/spam folders) Click on Verify my account Login with above credentials			
3	Faculty can view all applications to his university. Steps: Login with above credentials			
4	Faculty can filter by single research interest. Steps: Select a research interest from research interests dropdown. Click on Filter			
5	Faculty can filter by multiple research interests. Steps: Select **Multiple** from research interests dropdown. Check one or more research interests from pop up window. Click on Done. Click on Filter			
6	Faculty can filter by undergraduate university. Steps: Select an undergraduate university from undergrad university dropdown.			

	Click on Filter			
7	Faculty can filter by cumulative GPA. Steps: Slide the CGPA slider to a desired range Click on Filter			
8	Faculty can filter by GRE Quantitative Score. Steps: Slide the GRE Quantitative slider to a desired range Click on Filter			
9	Faculty can filter by GRE Verbal Score. Steps: Slide the GRE Verbal slider to a desired range Click on Filter			
10	Faculty can filter by PhD Degree Objective Score. Steps: Slide the PhD Degree Objective slider to a desired range Click on Filter			
11	Faculty can filter by MS Degree Objective Score. Steps: Slide the MS Degree Objective slider to a desired range Click on Filter			
12	Faculty can filter by any combination of above criteria. Steps: Select any combination of criteria from drop downs and sliders Click on Filter			
13	Faculty can sort applications by clicking on column headers Steps: Click on any column header			
14	Faculty can search applications by typing in Search field Steps: Type some letters in Search field			
15	Faculty can reset criteria by clicking on Reset Steps: Click on Reset			
16	Faculty can log out Steps: Click on username at top right corner, then click Log out			

Sign-off Procedure:

Comments:

(Mention comments for any tests with their test nos)

.....
Signature and Date

Prof. Hans Johnson
University of Iowa
External Product Owner

Point of Contact:

harsha-pitawela@uiowa.edu
Harsha Pitawela
Product Owner-Sprint 4
Gremester Project



GREMESTER_MAINRELEASE 2.0

User Acceptance Test Document

Title:

GREMESTER_MAINRELEASE 2.0

Objective:

The purpose of Gremester is to provide a platform for potential graduate students to find suitable computer science graduate programs and members of admission committees to find suitable candidates for their graduate programs. The end-users of the system are potential graduate students and members of admission committees of computer science graduate programs.

Scope:

Scenarios related to faculty login, application filtering and profile evaluation by faculty.

Introduction:

Requirements were gathered from users representing admission committees. Application filtering functionality has been implemented as per those requirements. In the UAT (User Acceptance Testing) phase, scenarios in this functionality will be tested and accepted/rejected by actual users. Test faculty account has been created for UAT.

Features to be Tested:

1. Faculty login
2. Application filtering by faculty on a range of criteria
3. View student's profile
4. View location(Google Map) of undergraduate college of the student
5. Evaluation of students by faculty
6. View evaluations done by faculty
7. View evaluations done by other faculty members

Approach:

Currently, seven mock student profiles have been created. Out of these profiles, six students have applied to University of Iowa and one student has applied to Arizona state university. Once all tests are performed, the development team will evaluate the test report and do bug fixes if necessary.

Test Environment Details:

Testing can be done by logging into <http://gremester.herokuapp.com>

Test faculty account credentials:
 E-mail: hans-johnson@uiowa.edu
 Password: 12345678

Tests:

#	Acceptance Testing Action	Success Criteria	Pass /Fail	Date	Initials
1	Faculty can login to the application Steps: Click Login > Faculty Login with above credentials	Faculty member should see a success message when logged in			
2	Faculty can view all applications to his university. Steps: Login with above credentials	Faculty members should see only six profiles once they are logged in since the test faculty account has been created for University of Iowa. Faculty member should not see the application of below student since he has applied to Arizona state university: Rahul Dravid			
3	Faculty can filter by single research interest. Steps: Select a research interest from research interests dropdown. Click on Filter	Faculty members should see only those profiles with the specified research interest.			
4	Faculty can filter by multiple research interests. Steps: Select **Multiple** from research interests dropdown. Check one or more research interests from pop up window. Click on Done. Click on Filter	Faculty members should see only those profiles with the specified research interests.			
5	Faculty can filter by undergraduate university. Steps: Select an undergraduate university from undergrad university dropdown. Click on Filter	Faculty members should see only those profiles with the specified undergraduate university.			
6	Faculty can filter by grading scales. Steps:	Faculty members should see only those profiles within the			

	Slide the CGPA slider to a desired range and select the grading scale from the drop down list Click on Filter	specified GPA range of Grading scale.			
7	Faculty can filter by GRE Quantitative Score. Steps: Slide the GRE Quantitative slider to a desired range Click on Filter	Faculty members should see only those profiles within the specified GRE quantitative score range.			
8	Faculty can filter by GRE Verbal Score. Steps: Slide the GRE Verbal slider to a desired range Click on Filter	Faculty members should see only those profiles within the specified GRE verbal score range.			
9	Faculty can filter by PhD Degree Objective Score. Steps: Slide the PhD Degree Objective slider to a desired range Click on Filter	Faculty members should see only those profiles within the specified PHD Degree Objective Score range.			
10	Faculty can filter by MS Degree Objective Score. Steps: Slide the MS Degree Objective slider to a desired range Click on Filter	Faculty members should see only those profiles within the specified MS Degree Objective Score range.			
11	Faculty can filter by Term. Steps: Select any term from the drop down list. Click on Filter	Faculty members should see only those profiles with the specified term.			
12	Faculty can filter by Year Steps: Select any year from the drop down list. Click on Filter	Faculty members should see only those profiles with the specified year.			
13	Faculty can filter by any combination of above criteria. Steps: Select any combination of criteria from drop downs and sliders Click on Filter	Faculty members should see only those profiles filtered by the combination of above criteria.			
14	Faculty can sort applications by clicking on column headers Steps:	Faculty members should see profiles with selected column			

	Click on any column header in the list: Name, Past Education, Research Area, GPA, GRE(Q), GRE(V)	being sorted by alphabet or by increasing order			
15	Faculty can search applications by typing in Search field Steps: Type some letters in Search field	Faculty members should see only those profiles filtered by the search text			
16	Faculty can reset criteria by clicking on Reset Steps: Click on Reset	Faculty members should see all filters being reset and all applications back.			
17	Faculty can log out Steps: Click on username at top right corner, then click Log out	Faculty member should see message confirming successful log out			
18	Faculty can view students profile Steps: Click on Student's name under the column header 'Name' on applications filtering page.	Faculty member should be able to view students profile			
19	Faculty can view the location(google map) of undergraduate college of the student. Steps: Click on Student's name under the column header 'Name' on applications filtering page. Click on undergraduate college link under the label 'Undergraduate School:'	Faculty member should be able to see the location(google map) of undergraduate college of the student			
20	Faculty can evaluate a student's profile Steps: Click on Student's name under the column header 'Name' on applications filtering page. Click on Add Evaluation button. Select application score and ee_background from the drop down list respectively. Add comments in the comments box and click 'Submit'	Faculty member should be able to submit evaluation successfully. Faculty member should see the message 'Your evaluation has been submitted!' on the student's profile view page.			
21	Faculty can view their own evaluations done so far. Steps:	Faculty members should be able to see their own evaluations with the below information:			

	Click on username at top right corner, then click on 'View Evaluations'	Student Name Email Applied Term Date of Evaluation Score Comments Evaluation			
22	Faculty can view other faculty evaluations Steps: Click on username at top right corner, then click View Evaluations Click on 'View other evaluations' under the column header 'Evaluation' for each of the student	Faculty should be able to see evaluations of other faculty members for each student if any.			

Sign-off Procedure:

Comments:

(Mention comments for any tests with their test nos)

.....
Signature and Date

Prof. Hans Johnson

University of Iowa

External Product Owner

Point of Contact:

avantiharish-deshmukh@uiowa.edu

Avanti Deshmukh

Product Owner-Sprint 5

Gremester Project

Front end technology change

We have decided to stop using Angular as the main front end framework for our project. After two weeks of learning and trying to integrate Angular into our project, the team has realized that the lack of documentation and incompatibility between Angular and Rails will block our team progress to deliver MVP by the end of the semester. Instead of using Angular, we have decided to use Haml/Html/Bootstrap and Javascript for our client.

Type of change

Change of front end technology (AngularJs)

What has triggered the change?

During the first week of Sprint 1, we have been working on creating user interface for user login and registration in Angular (Figure 1,2,3). However, we are struggling to seamlessly integrate AngularJs with Rails. Some of the gems that we are using for user authentication, i.e Devise and Rails-admins, doesn't work well with Angular. Even with tutorials on how to set up these Gems with Angular, we were having difficulty understanding how data is transferred between Rails and Angular. Similarly, routing becomes more complex when Angular utilizes different routing mechanism from Rails. At the same time, there is not much documentation on how we can connect these two technologies together. Because of such independence and lack of configuration/documentation, the current application is more prone to bug and its quality will be compromised. More importantly, we're worried that even when we are familiar with the current technology, the speed of product delivery won't improve significantly because of such isolation. Given the pace of development for the first sprint, we will not be able to deliver the product that we already planned by the end of sprint 1, which can potentially impact our deliverables by the end of the course. This problem with Angular leaves us with 3 main options to consider:

1. Continue working with Angular but narrow down the scope of the project
2. Change the technology to one that the team more familiar with and that integrates better with Rails without changing the scope of the project
3. Switch to a completely new technology framework where each components integrate well with each other

To make a decision on which option to proceed, we were required to re-evaluate our priorities - whether we want to 1) proceed to learn new technology stack and lower the complexity of product engineering or 2) to use the technology stack that we are already familiar with and implement the product that is more engineering heavy. The team reached a consensus that we should focus on the later goal: utilizing the tools that we know better to allow more room for product improvement. Undeniably, the fact of choosing the stack we already know will limit us

from learning a new development framework. However, it doesn't mean we are limiting our learning opportunities. Given the comparative complexity of the project (compared to our past projects), we will have to pick up different technologies to solve our problem. With that, we decided to go with option 2

The impact on the project

When we decided to use Angular initially, we wanted to adopt a component-based architecture for the front end to account for the performance problem of Ruby. Hence, changing the front end technology will definitely impact the speed of the project in the long run. It was our team oversight when we came to our decision to use Angular without much research. However since it is early in our development, this change will allow us to have more time to optimize through database caching, creating reusable UI components and improving user experience through JavaScript. For sprint 1, we can revert our code that has the angular modulars without much issue. The impact will be that we will need to do extra work to get this sprint back on track after having to remove all the angular components and rewriting them with devise/haml. We will also need to do revise our test plan because of the change in tools. Otherwise, we will follow with our initial project scope. Tools for TDD, BDD and group collaboration protocol won't be impacted.

Revised strategy to deliver MVP for Sprint 1

Our MVP for Sprint 1 is the application with successful login/logout and sign up for faculty and student. In order to accomplish this goal, we will have to revert the angular changes from master. We will use devise and rails-admin gem for user authentication and admin UI. Everyone is responsible for writing tests for their own feature. We will have to double the work over the weekend to account for the change. Regarding our backlog tasks, issue 40 and 44 will no longer relevant since it was related to Jasmine test for Angular code. Therefore, the adjusted velocity points for this sprint is 38 instead of 42 initially. Work will be divided as follows:

- Julia will work on front-end for login and signup
- Linh will work on the workflow between faculty and admin
- Avanti will on controllers for form validation
- Harsha will work on models for faculty and admin

Resources

Even though we decided to move on with Angular, we all agree it's a good framework for frontend development. For anyone inheriting this project and looking into Angular to migrate to, here are some resources we found useful. All our Angular-related code will be in angular branch on our GitHub project page for reference.

<https://medium.com/@avatsaev/angular-2-and-ruby-on-rails-user-authentication-fde230ddaed8>

<https://thinkster.io/tutorials/angular-rails>

Gremester - User Guide

This document guides users on how to perform all important features in Gremester web app.

User - Sign Up

User Sign Up - Student

In the Gremester homepage, go to Sign Up > Student.

Enter the required information in the following form and click on “Sign up”.



Sign Up

Username

First name

Last name

Email

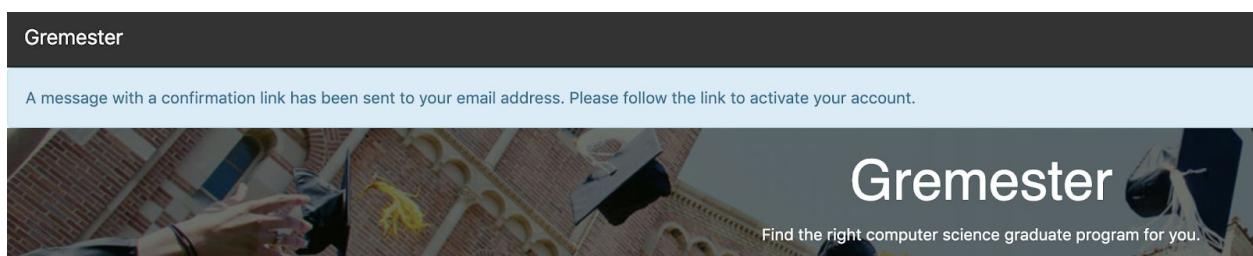
Password

Password confirmation

[Log in](#)

[Didn't receive confirmation instructions?](#)

After successful sign up, you will get an alert similar to the following.



Now you need to go to your email account to activate the account. Rarely the confirmation email could be in your junk/spam folder.

Click on “Confirm my account” to activate your account.

[External] Gremester confirmation instructions

G

gremestersep2019@gmail.com

Mon 4/29/2019 12:01 PM

Ekanayake Wasala Mudiyanselage, Harsha Rakkitha Bandara Pitawela ✎

Welcome Harsha!

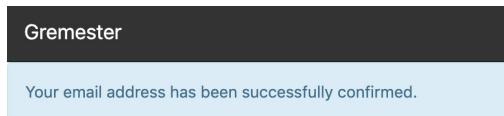
You can confirm your account email through the link below:

[Confirm my account](#)

Thanks,

Gremester Services

Once you click on “Confirm my account”, you will be redirected to Gremester Student Login, where you could enter your email and password to login.



Log in

Email

Password

Remember me

[Log in](#)

[Sign up](#)

[Forgot your password?](#)

[Didn't receive confirmation instructions?](#)

User Sign Up - Faculty

In the Gremester homepage, go to Sign Up > Faculty.

Enter the required information in the following form and click on “Sign up”.

Please note that you will need to provide a university issued ID card or a university faculty web page to register as faculty.



Sign Up

Username

First name

Last name

Email

Password

Password confirmation

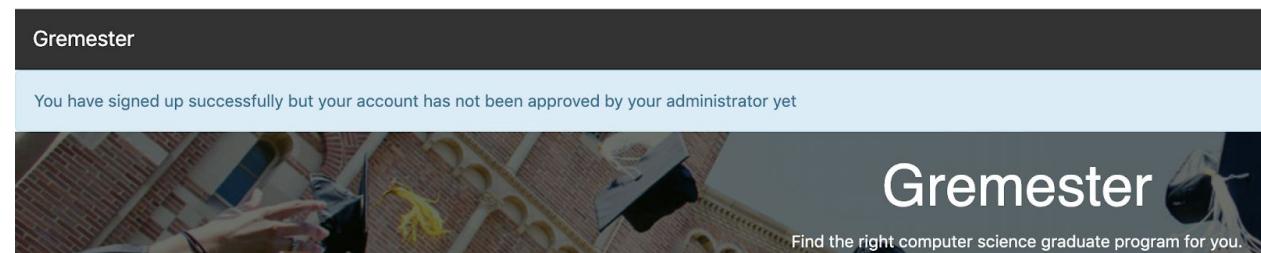
University
 Arizona State University ▾

Id card
 Choose File | No file chosen

Link to Faculty Website

[Log in](#)
[Didn't receive confirmation instructions?](#)

After successful sign up, you will get an alert similar to the following.



Once the administrator approves your account, faculty could log in using the faculty login.

User Login

User Login - Student

In the Gremester homepage, go to Login > Student.

Enter your email and password in the following form and click on “Log in”.



Log in

Email

Password

Remember me

Log in

[Sign up](#)

[Forgot your password?](#)

[Didn't receive confirmation instructions?](#)

Upon successful login, you will be on the Gremester Student homepage.

A screenshot of the Gremester Student homepage. At the top, there is a dark header bar with the "Gremester" logo on the left and a user profile "Hi, harsha3" on the right. Below the header, a light blue banner says "You are logged in". The main content area has a title "Graduate Schools and Programs". It includes search and filter options: "Type All", "Ranking from 1 To 10", a "Filter" button, "Show 10 entries", and a "Search:" input field. A table lists graduate schools with columns for Rank, Name, Public/Private, Acceptance Rate, Tuition, Website, and Action (with a "Show" link).

Rank	Name	Public/Private	Acceptance Rate	Tuition	Website	Action
1	Massachusetts Institute of Technology	Private	18.0	\$49,600	http://www.mit.edu/	Show
2	Stanford University	Private	17.0	\$16,900	http://www.stanford.edu/	Show
3	University of California – Berkeley	Public	14.0	\$28,000	https://www.berkeley.edu/	Show
A	California Institute Of Technology	Private	0.0	\$48,200	http://www.caltech.edu/	Show

User Login - Faculty

In the Gremester homepage, go to Login > Faculty.

Enter your email and password in the following form and click on "Log in".



Log in

Email

Password

Remember me

Log in

[Sign up](#)

[Forgot your password?](#)

[Didn't receive confirmation instructions?](#)

Upon successful login, you will be on the Gremester Faculty homepage.

The screenshot shows the Gremester Faculty homepage. At the top, there is a dark header bar with the 'Gremester' logo and a user greeting 'Hi, harsha_fac~'. Below the header, a light blue banner displays the message 'You are logged in' and the name 'Professor Harsha Ekanayake, University of Iowa'. The main content area features several search filters: 'Research Interests' (dropdown), 'Cumulative GPA: 0 - 100 %' (range slider), 'Scale: **All**' (dropdown), 'GRE-Quantitative range: 130 - 170' (range slider), 'MS Degree Objective range: 0 - 5' (range slider), 'Undergraduate College' (dropdown), 'GRE-Verbal range: 130 - 170' (range slider), 'PhD Degree Objective range: 0 - 5' (range slider), 'Term' (dropdown), and 'Year' (dropdown). Below the filters are two buttons: 'Filter' (blue) and 'Reset' (orange). A 'Show' dropdown set to '10 entries' and a 'Search:' input field are also present. The main content area contains a table with columns: Name, Past Education, Research Area, GPA, GRE(Q), GRE(V), and MS/PhD. The table lists four entries:

Name	Past Education	Research Area	GPA	GRE(Q)	GRE(V)	MS/PhD
Harsha Pitawela	University of Iowa	Artificial Intelligence, Augmented & Virtual Reality, Data Science	3.56	164	146	0/0
Dwayne Wade	University of California – Berkeley	Artificial Intelligence, Data Management & Visualization, Fabrication	2.5	164	146	2/4
Kevin Durant	University of Texas at Arlington	Artificial Intelligence, Molecular Information Systems	4.0	164	146	4/2
Tim Duncan	Georgia Institute of Technology	Artificial Intelligence, Ubiquitous Computing	2.1	164	146	1/3

User Login - Admin

Visit the admin login page through gremester.herokuapp.com/admins/sign_in. Enter admin email and password in the following form and click on “Log in”.



Log in

Email

Password

Remember me

Log in

[Sign up](#)
[Forgot your password?](#)

Upon successful login, admin should see the Gremester Admin Dashboard.

Model name	Last created	Records
Admins	6 days ago	2
Applications	7 days ago	7
Categories	8 days ago	0
Countries	8 days ago	249
Faculties	about 1 hour ago	3
Faculty evaluations	6 days ago	2

Student Features

Edit Profile

Upon successful login as a student, click on username(at top right corner) and then click on “View Profile.”

The screenshot shows the Gremester Profile page. On the left sidebar, there are three options: "My Profile", "My Applications", and "Admission Predictions". The main content area is titled "Profile". It features a placeholder profile picture and a blue "Edit Profile" button. To the right, under "Personal info", there are several input fields: Name (Harsha Pitawela), Gender (Male), Citizenship (Sri Lanka), Degree Objective Phd (5), Degree Objective Master (5), Research Interest (Artificial Intelligence, Augmented & Virtual Reality, Data Science), and Undergraduate School (Iowa State University, B.S. Computer Science, 2009 - 2013, GPA: 3.56, Standard with A+).

Click on button “Edit Profile”.

Fill in the profile details and click on “Save Changes” button.

University Applications

In the View Profile screen, click on “My Applications” from the left navigation bar.

The screenshot shows the Gremester "My Applications" page. The left sidebar has the same three options: "My Profile", "My Applications", and "Admission Predictions". The main content area is titled "My Applications". It displays a table with one row of data: University Name (University of Iowa), Applied Date (04/21/2019), Admit Date (Spring 2020), Reject Date (N/A), Interested Term/Year (Spring 2020), Delete Application (yellow bin icon), and Edit Application (green edit icon). Below the table is a "Add School" button.

If you want to update or remove any applications, click on either green edit button or yellow bin button respectively.

If you want to add a new university application click on the button “Add School” and fill in the fields in the following form.

Gremester

My Profile
My Applications
Admission Predictions

My Applications

University Name	Applied Date	Admit Date	Reject Date	Interested Term/Year	Delete Application	Edit Application
University of Iowa	04/21/2019			Spring 2020		
University Name	Application Status	Date	Interested Term	Interested Year		
select un ▾	select status ▾	mm/dd/yyyy ▾	select term ▾	select year ▾		

Add School

Admission Predictions

In the View Profile screen, click on “Admission Predictions” from the left navigation bar. Select university from the list and click on the button “Calculate”.

Gremester

My Profile
My Applications
Admission Predictions

Calculate Admission Chance

University Name	Admission Category	Admission Chance
select university		

You will see the admission chance calculated by Gremester for your selected university.

Gremester

My Profile
My Applications
Admission Predictions

Calculate Admission Chance

University Name	Admission Category	Admission Chance
select university		
University of Iowa	Target	48%

Discussion Forums

Upon successful login as a student, click on username(at top right corner) and then click on “Discussion Forum.”

Search Topics and Posts

Notification Settings harsha Main Page Sign Out

All Messageboards

Questions about CS Master Program in US

0 posts in 0 topics

Posts/Questions for any specific programs in Computer Science

Click on any category to start a discussion topic.
Fill in details and click “Create New Topic.”

Search Topics and Posts

Notification Settings harsha Main Page Sign Out

All Messageboards > Questions about CS Master Program in US

Title

Start a New Topic

Content

Create New Topic

Your topic will then be displayed in the said category.

Search Topics and Posts

Notification Settings harsha Main Page Sign Out

All Messageboards > Questions about CS Master Program in US

Start a New Topic

1

Funding 1 minute ago harsha

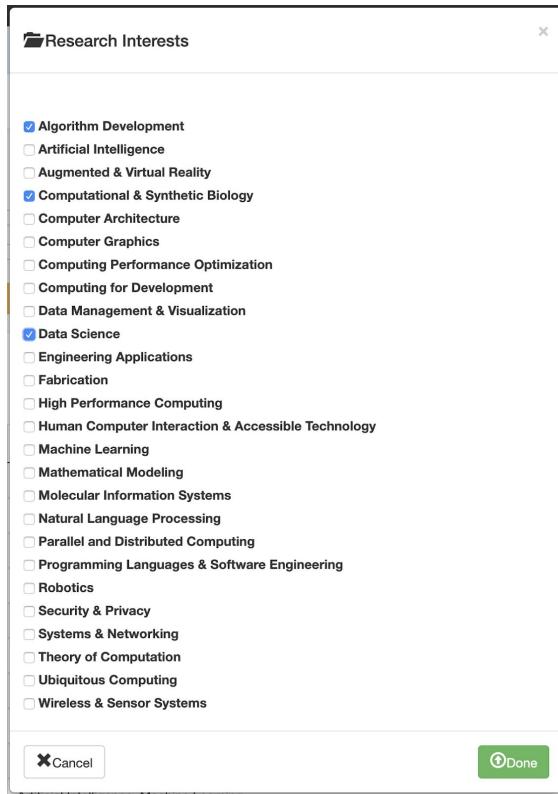


Faculty Features

Filter Applications

Upon successful login as a faculty, you can select any criteria to filter applications.

Important: If you want to select multiple research interests, select “**Multiple**” from Research Interests dropdown, then select multiple interests from the list and click on “Done” button.



After you have selected necessary filter criteria, click on “Filter” button and filtered applications will be displayed.

If you want to remove all filters and get the complete list of applications, click on “Reset” button.

View Applicant Profile

Click on the name field of an application to view the full profile of the applicant.

[Back](#)*Evaluation for this application has not been submitted*[Add Evaluation](#)**Harsha Pitawela's Profile**

Gender:	<input type="text" value="Male"/>
Citizenship:	<input type="text"/>
Degree Objective Phd:	<input type="text" value="0"/>
Degree Objective Master:	<input type="text" value="0"/>
Research Interest:	<input type="text" value="Artificial Intelligence, Augmented & Virtual Reality, Data Science"/>
Undergraduate School:	University of Iowa B.S. Computer Science 2010 - 2014 GPA: 3.56, Standard with A+
TOEFL Score:	<input type="text" value="110"/>
GRE Writing:	<input type="text" value="4.0"/>
GRE Quantitative:	<input type="text" value="164"/>
GRE Verbal:	<input type="text" value="146"/>
Graduate Start Term:	<input type="text" value="Fall 2019"/>
Work Experience:	<input type="text" value="4 - 5 years"/>
SOP:	<input type="text"/>

Add Evaluation

Follow steps to view applicant profile as indicated in the above section and click on “Add Evaluation” button.

[Back](#)*Evaluation for this application has not*[Add Evaluation](#)**Harsha Pitawela's Profile**

Fill in the form below and click “Submit” to save your evaluation.

[Back to profile](#)

Application Review for Harsha Pitawela

Applicant Score

EE Background

Comments

Submit

View My Evaluations

Upon successful login as a student, click on username(at top right corner) and then click on “View Evaluations.”

My Evaluations

Show
10 entries

Student Name	Email	Applied Term	Date Of Evaluation	Score	Comments	Evaluation
Harsha Pitawela	harshainfo@gmail.com	Spring 2020	04-23-2019	4	Good Profile. I would recommend for admission.	View other evaluations
Avanti Deshmukh	avanti.deshmukh532@gmail.com	Fall 2019	04-23-2019	3	Conditional admit as it lacks EE background.	View other evaluations

Showing 1 to 2 of 2 entries

[Previous](#) **1** [Next](#)

View Other Faculty Evaluations

In the above, My Evaluations page, click on “View other evaluations”.

If there are any evaluation by other faculty members, they will be displayed here.

Other Evaluations

Show
10 entries

Professor Name	Date of Evaluation	Score	Comments
No data available in table			

Showing 0 to 0 of 0 entries

[Back](#)

Admin Features

Faculty Approval

In the admin dashboard, click on “Faculties”.

The screenshot shows the Gremester Admin Site Administration dashboard. On the left, a sidebar lists various administrative categories: NAVIGATION (Admins, Applications, Countries, Faculties, Faculty evaluations, Grading scales, Grading scale types, Profiles, Rank types, Rankings, Research interests, Students, Thredded messageboards), and a central Dashboard link. The main area is titled "Site Administration" and displays a message "You are logged in". Below this is a "Dashboard" button with a house icon. A table lists the number of records for different models, with a "Faculties" row highlighted. The table has columns for Model name, Last created, and Records. The "Faculties" row shows "about 7 hours ago" and "3" records. At the bottom of the table, there are navigation links for "Previous" and "Next".

Model name	Last created	Records
Admins	6 days ago	2
Applications	7 days ago	7
Categories		0
Countries	8 days ago	249
<u>Faculties</u>	about 7 hours ago	3

Click on the ✓ mark at the right end of the row, which contains details of the faculty member that you want to approve.

List of Faculties

The screenshot shows a list of three faculties:

<input type="checkbox"/> First name	Last name	Email	Weblink	University	Username	...
<input type="checkbox"/> Harsha	Ekanayake	harsha-pitawela@uiowa.edu	http://myweb.uiowa.edu/hpitawela/	University #61	harsha_fac2	... ✓ i x
<input type="checkbox"/> Hans	Johnson	hans-johnson@uiowa.edu	https://www.engineering.uiowa.edu/facul...	University #61	Hans	... i x
<input type="checkbox"/> Harsha	Pitawela	hpitawela@uiowa.edu	http://myweb.uiowa.edu/hpitawela/	University #61	harsha_fac	... i x

3 faculties