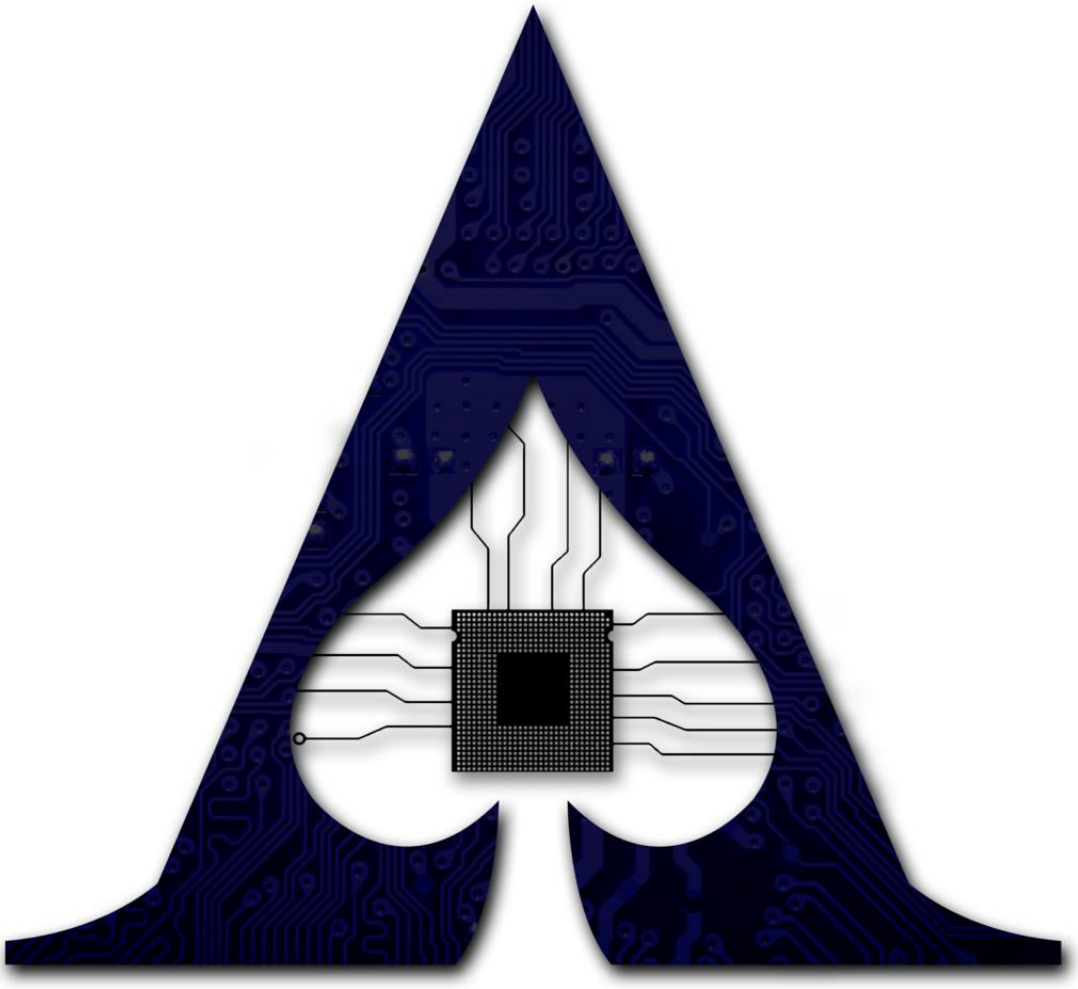
Final Deliverable

Ace of Spades



Diana Soto, Gerrit Steinbach, Bar Slutsky,
Kezaram Tharmasegaram, Jonathan Jung-Kyun An

Product Backlog (Admin & Student)

- ✓ As Nate/Sohee**, I** would like to be able to Login to the admin part of the application
- ✓ As Nate/Sohee, I would like to be able to add problems
- ✓ As Nate/Sohee, I would like to be able to remove problems
- ✓ As Nate/Sohee, I would like to be able to edit problems
- ✓ As Nate/Sohee, I would like to be able to add users to the software
- ✓ As Nate/Sohee, I would like to be able to edit existing users in the software
- ✓ As Nate/Sohee, I would like to be able to delete users from the software
- ✓ As Nate/Sohee, I would like to be able to create a assignment.
- ✓ As Nate/Sohee I would like to be able to set a due date for an assignment.
- ✓ As Nate/Sohee, I would like to be able to view all and individual student results on specific assignments.
- ✓ As Nate/Sohee, I would like to be able to view a student's grade.
- ✓ As Nate/Sohee, I would like to be able to sort student assignment results by grade, student uid, and number of attempts for an assignment.
- ✓ As Nate/Sohee, I want to be able to see a leaderboard of my student sorted by highest to lowest overall grade then, in case of a tie, lowest to highest completion time."
- ✓
- ✓ As Sohee I would like to be able to change due date for an assignment.
- ✓ As Brian, I would like to be able to Login as a student to the application
- ✓ As Brian, I would like to be able to view posted assignments
- ✓ As Brian, I would like to be able to complete an assignment
- ✓ As Brian, I would like to be able to receive immediate results for my submission
- ✓ As Brian, I would like to be able to view info about assignments (e.g. due dates and progress)
- ✓ As Brian, I would like to be able to save my progress in the middle of work, so that I can close the app and come back to it later.
- ✓ As Brian, I would like to be able to review my entire work after submission.
- ✓ As Brian I want to be able to have a list of all my past attempts for any particular assignment so I can view my grade, submission date and a button option to view attempt progress"
- ✓ As Nate/Sohee I would like to be able to edit a student's assignment mark.
- ✓ As an admin, I want to be able to add new problems using the latex syntax so that I can visually represent complicated math equations/formulas.

- ✓ As a student, I want to be able to see an option to export any current attempt problems into a latex pdf, such that students can work on assginments offline and without technological restrictions.
- ✓ As an admin, I want to be able to see a graphical representation of the leaderboard.
- ✓ As a student, I want to be able to use able to a the leaderboard containing marks of all students in order from highest to lowest grade then by lowest to highest time.
- ✓ As an admin, I want to be able to input a problem format, and have the application automatically generate problems for me that are based on that format.
- ✓ As an admin, I want to be able to add a hints feature into a problem so that I can guide students on the right path when answering a question.
- ✓ As a student, I want to be able to view a hints feature into a problem so that I can be led on the right path when attempting a problem.

Sprint #3: October 30th - November 6th

Story 1: (Total = 7 points)

"As an admin, Sohee, I want to have a screen where I can enter details about a new assignment and have the assignment created for me in the system, so that I can assign it to students whenever I want."

- Task 7: Create the GUI for adding a new assignment. This screen will include entry boxes for each field of the assignment class: assignment_id, assignment_name, generating_formula, assignment_deadling, visibility.
  (Diana, Gerrit) - 2 points

✓ Task 8: Implement API functions to for the functionality required by the user story. (Bar) - 3 points

- Task 9: Integrate GUI with API functions such that the functionality described in the user story is working as expected.
  Dependencies: T1, T2
  (Bar, Diana, Gerrit) - 2 points

Story 2: (Total = 7 points)

"As a Student , I want to be able to view a list of all the assignments that were assigned to me, so that I can access each assignment."

- Task 10: Create the GUI for viewing a student's assigned assignments. This screen will include a list of rows, each one describing an assignment's name, deadline and grade, followed by a "view previous attempts" button and a "new attempt" button
  (Diana, Gerrit) - 3 points

✓ Task 11: Implement API functions to for the functionality required by the user story.
(Bar) - 2 points

✓ Task 12: Integrate GUI with API functions such that the functionality described in the user story is working as expected.
Dependencies: T1, T2
(Bar, Diana, Gerrit) - 2 points

Story 3: (Total = 8 points)

"As a Student, I want to have a screen for each assignment, in which I can view the assigned questions, input my answer, save my progress and submit it when I'm done"

- Task 13: Create the GUI for viewing a specific assignment. This screen will display a list of all the assigned questions for that assignment, with an entry box for each

corresponding answer. It will have a button for saving progress and another button for final submission.
(Jon, Kezaram) - 3 points

- Task 14: Implement API functions to for the functionality required by the user story.(Jon, Kezaram) - 3 points

- ✓ Task 15: Integrate GUI with API functions such that the functionality described in the user story is working as expected.
Dependencies: T1, T2
(Jon, Kezaram) - 2 points

Sprint #4: November 7th - November 12th

Story 1: (Total = 6 points) Kezaram

“As a student, I want to be able to have a list of all my past attempts for any particular assignment so I can view my grade, submission date and a button option to view attempt progress”

- Task 1: (2 points)
  Retrieve grade and submission date for each attempt from specific assignment table according to the assignment button the user clicked.

- Task 2: (2 points)
  Create Date of Submission + Grade labels and View Attempt buttons dynamically on GUI. This was based of the number of attempts the user had for a particular assignment.
  – Dependency T1

- Task 3: (2 points)
  Connect the GUI to data from database. When the user clicks the “Past Attempts” button, their screens should be switched to a screen that creates the labels and view buttons automatically.
  –Dependency T1 & T2

Story 2: (Total = 11 points) Jon

“ As an admin, I want to be able to see a leaderboard of my student sorted by highest to lowest overall grade then, in case of a tie, lowest to highest completion time”

- Task 1: (1 points)
  Create functionality that retrieve a list of all students with the following order:
  -The highest to lowest grade
  -In case of a tie in grades, lowest to highest overall completion time.
  Update user and user’s table to have “grade” and “time” columns.

- Task 2: (2 points)
  -Create a screen for the leaderboard that will dynamically create entries for every student in the database.

- Task 3: (2 points)
  -Create functionality that calculates the number of seconds passed between two points in time according to the date/time format used in the application.

- Task 4: (3 points)

-Calculate user's total time and average grade for every latest submission for all assignments.
–Dependency T3

- Task 5: (1 point)
  Every time student makes a new submission, update users' time and grade in the database.
  –Dependency T3 & T4

- Task 6: (2 points)
  Connect GUI to the database functionality. When user navigates to Leaderboards screen, it should display a list of users by the
  –Dependency on T1->T5

Story 3: (Total = 7 points) Bar

"As a student, I want to be able to work on an assignment by inputting my solution and having the ability to both save my progress and submit for marking"

- Task1: (0.5 points)
  add "save" and "submit" buttons to GUI

- Task2: (2.5 points)
  create function for save button: function will fetch the texts from all the entries in the frame, (make all entries into list so can iterate and get all texts) then it will create a string in format: 'ans1','ans2',...and store that string into the attempt row in the a# table for that user

- Task 3: (3 points)
  write a function to compare each answer to the one in the database and create an algorithm for calculating the grade.
  create submitting function which will update the progress(task1), grade and submission time columns in the a# table

- Task 4: (0.5 points)
  make the row generation function update the entries with existing progress upon creation

- Task 5: (0.5 points)
  create the links between the buttons in the "view user attempts" menu and the corresponding functions to display the correct new screen

Story 4: (Total = 3 points) Bar

As a student, I want to be able to see a specific attempt for an assignment.
–Dependency S1

- Task 1: (2.5 points)
  Create a class based on the view attempt class, which will have labels instead of entries It will also include the functionality for submission and progress saving.

- Task 2: (0.5 points)
  create the links between the buttons in the “view past attempts” menu and the corresponding functions to display the correct new screen

Sprint #5: November 13th - November 20th

S2 Continued (6 points)- Jon

"As an admin, I want to be able to see a leaderboard of my student sorted by highest to lowest overall grade then, in case of a tie, lowest to highest completion time."

- Task 1: (3 points)
  Calculate user's total time and average grade for every latest submission for all assignments.
- Task 2 : (1 point)
  Every time student makes a new submission, update users' time and grade in the database.
  –Dependency T1
- Task 3 : (2 points)
  Connect GUI to the database functionality. When user navigates to Leaderboards screen, it should display a list of users by the
  –Dependency T1 & T2

Story 5: (Total= 8.5) Diana + Gerrit

"As an admin, I want to be able to see all student grades and progress for any given assignment"

- Task 1: (0.5 points)
  Create a class for ViewStudentGrades which is where the GUI screen will be coded and frame to main.py

- Task 2: (2 points)
  Create database function to get users that have worked on an assignment

- Task 3: (6 points)
  Code GUI labels, buttons, entries, listbox and dropdown to work with the database functions to show the admin grades and progress for a selected assignment from the dropdown
  –Dependency Story T1 & T2

Story 6: (Total=) Gerrit + Diana

"As an admin, I want to be able to filter the results from the previous screen according to username."

- Task 1: (1 point)
  Add filter, and sort buttons/labels to the GUI from S5
  –Dependency Story 5

- Task 2: (3 points)
  Update the existing table created in T1

Create a clear filter method that restores the table to its original state
Create methods to see

- Task 3: (2 points)
  Connect methods to GUI for admin to edit see filtered results in the listbox without switching screens, update the database and the screen after changes

Testing

Test "Manage Problems" Cases (3 points)- Kezaram

- Task 1 : TestProblem
  - test_init = Initialize extreme cases of a Problem Object
  - test_get_qid = Tests if correct value and type returned
  - test_get_subject = Tests if correct value and type returned
  - test_get_question = Tests if correct value and type returned
  - test_get_answer = Tests if correct value and type returned
  -
- Task 2: TestAddingProblem
  - test_empty_subject_add = Tests if a problem's subject can be added with empty string
  - test_empty_question_add = Tests if a problem's question can be added with empty string
  - test_empty_answer_add = Tests if a problem's answer can be added with empty string
  - test_add_valid_answer = Tests if a valid problem can be added

- Task 3: Test RemovingProblem
  - test_delete_problem = Tests if valid problem can be deleted
  - test_delete_non_existant_problem = Tests if invalid problem can be deleted

- Task 4: TestUpdatingProblem
  - test_empty_subject_update = Tests if a problem's subject can be updated with empty string
  - test_empty_question_update = Tests if a problem's question can be updated with empty string
  - test_empty_answer_update = Tests is a problem's answer can be updated with empty string
  - test_update_non_existant_problem = Tests if a invalid problem can be updated

Test "View User Assignment" Cases (3 points) Kezaram

- Task 1: TestViewUserAssignments
  - test_view_assign_details = Tests if correct assignment details returned
  - test_view_assign_name_type = Tests if correct type returned
  - test_view_assign_formula_type = Tests if correct type returned
  - test_view_assign_deadline_type = Tests if correct type returned
  - test_view_assign_visibility_type = Tests if correct type returned
  - test_view_non_existant_assign = Tests if invalid assignment can be viewed

- Task 2: TestAddUserAttempts
  - test_add_attempt = Tests if a new attempt can be added

  - test_add_empty_attempt = Tests if an empty attempt can be added
  - test_duplicate_submission_dates = Tests if submission dates are duplicated
  - test_add_attempts_for_non_existant_assign = Tests if invalid assignment attempt can be added

- Task 3: TestViewUserAttempts
- test_get_user_attempts = Tests if correct user attempts are retrieved
- test_get_non_existant_nth_attempt = Tests if invalid nth attempt can be retrieved
- test_get_user_nth_attempt = Tests if valid nth attempt can be retrieved
- test_get_user_assignment_progress = Tests if correct progress info of assignment retrieved
- test_view_graded_submission_upon_submit = Tests if grade and submission date co-exist

Test "Login" Cases (2 points) Bar

- test_email_blank – Try logging in with blank email
- test_email_not_in_system – Try logging in with email that is not stored in the database
- test_password_blank – Try logging in with a blank password
- test_password_wrong – Try logging in with the wrong password for an existing email
- test_both_email_password_blank – Try logging in with both fields blank
- test_correct_combo_admin – Try logging in with the right credentials for an admin user
- test_correct_combo_student – Try logging in with the right credentials for a student user

Test "Add User" Cases (3 points) Bar

- test_empty_role_add – Try adding a new user with a blank role
- test_empty_name_add – Try adding a new user with a blank name
- test_empty_email_add – Try adding a new user with a blank email address
- test_empty_role_update – Try updating a user's details with a blank role
- test_empty_name_update – Try updating a user's details with a blank name
- test_empty_email_update – Try updating a user's details with a blank email address
- test_invalid_role_add – Try adding a new user with a role other than student or admin
- test_invalid_role_update - Try updating a user's role to a new role other than student or admin
- test_delete_user – Try deleting a user
- test_add_user – Try adding a user with valid information

Test "Create Assignment" Cases (3 points) Bar

- test_empty_name – Try creating a new assignment with an empty name
- test_empty_deadline – Try creating a new assignment with an empty deadline
- test_empty_visible – Try creating a new assignment with an empty visibility indicator
- test_empty_subject – Try adding a subject row to the assignment without indicating a subject
- test_empty_number_questions – Try adding a subject row to the assignment without indicating a number of questions

- test_non_natural_number_questions – Try adding a subject row to the assignment with input that is not a positive integer
- test_too_many_number_questions – Try adding a subject row to the assignment with a number of questions that is greater than the available number of questions for that subject
- test_non_existing_subject – Try adding a subject row to the assignment with a subject that is not associated with any problem in the database
- test_assignment_name_taken – Try creating a new assignment with a name that already
- exist in the database

Sprint #6: November 20th - November 24th

Story 1: (Total = 3 points)

As an admin, I want to be able to add new problems using the latex syntax so that I can visually represent complicated math equations/formulas.

Task 1: Create a function to convert problem strings into latex representation.

(Bar) - 2 points

Task 2: Display the converted latex representation when a student attempts an assignment.
Dependencies: T1
(Bar) - 1 point

Story 2: (Total = 8 points)

As a student, I want to be able to see an option to export any current attempt problems into a latex pdf, such that students can work on assginments offline and without technological restrictions.

- Task 3: Create the button the user can call the generate pdf class.
  (Kezaram) - 1 points

✓ Task 4: Add a function to convert latex string into a latex .tex file.
(Kezaram) - 2 points

✓ Task 5: Add a function to convert generated latex .tex file into a pdf file.
Dependencies: T4
(Kezaram) - 3 points

✓ Task 6: Integrate GUI back button with generating .tex and .pdf files.
Dependencies: T3, T4, T5
(Kezaram) - 2 points

Story 3: (Total = 7 points)

As an admin, I want to be able to see a graphical representation of the leaderboard.

- Task 7: Implement line graph functionality into the leaderboard
  (Jon) - 3 points

- ✓ Task 8: Update graph every time student makes new assignment submission
  (Jon) - 2 points

- ✓ Task 9: Properly lay out the graph onto the GUI
  Dependencies: T7, T8
  (Jon) - 2 points

Story 4: (Total = 2 points)

As a student, I want to be able to use able to a the leaderboard containing marks of all students in order from highest to lowest grade then by lowest to highest time.

- Task 10: Connect student menu to leaderboard screen
  (Jon) - 2 points

Story 5: (Total = 7 points)

As an admin, I want to be able to input a problem format, and have the application automatically generate problems for me that are based on that format.

- Task 11: Write the functions to generate random problems based on inout format
  (Gerrit) - 3 points

- ✓ Task 12: Implement API functions to for the functionality required by the user story.
  (Gerrit) - 2 points

- ✓ Task 13: Integrate GUI with API functions such that the functionality described in the user story is working as expected.
  Dependencies: T11, T12
  (Gerrit) - 2 points

Story 6: (Total = 7 points)

As an admin, I want to be able to add a hints feature into a problem so that I can guide students on the right path when answering a question.

Task 14: Create the GUI to add a hint field when adding a new problem.
(Diana) - 3 points

Task 15: Add database function to add the hint enetered into database.
Dependencies: T14
(Diana) - 2 points

Task 16: Integrate GUI with API functions such that when the admin enters a hint or the appropriate interface is shown.
Dependencies: T14, T15
(Diana) - 2 points

Story 7: (Total = 7 points)

As a student, I want to be able to view a hints feature into a problem so that I can be led on the right path when attempting a problem.

Task 17: Add hint button for every problem for the user to click on.
(Diana) - 3 points

Task 18: Add database function to add the hint enetered into database.
(Diana) - 2 points

Task 19: Integrate GUI with API functions such that when the student views a hint or the appropriate interface is shown.
Dependencies: T17, T18
(Diana) - 2 points