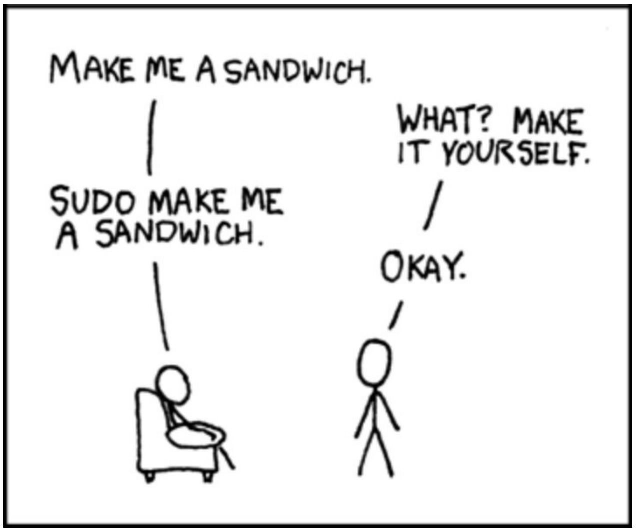


SUMMIT ANALYTICS

BT3103 Team 2 Documentation

—

The Team

Tay Jia Hui (Candice)

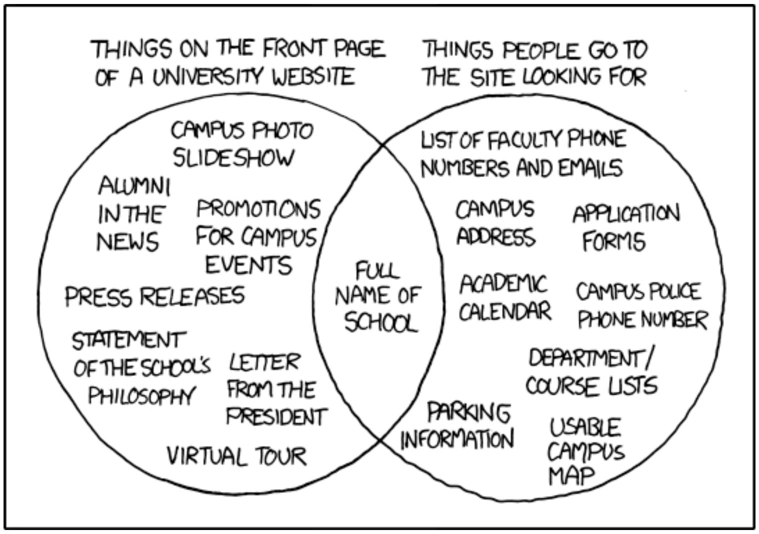
Li Xuan Guang (Arthur)

Suen Wai Lun

Ng Kai Wen

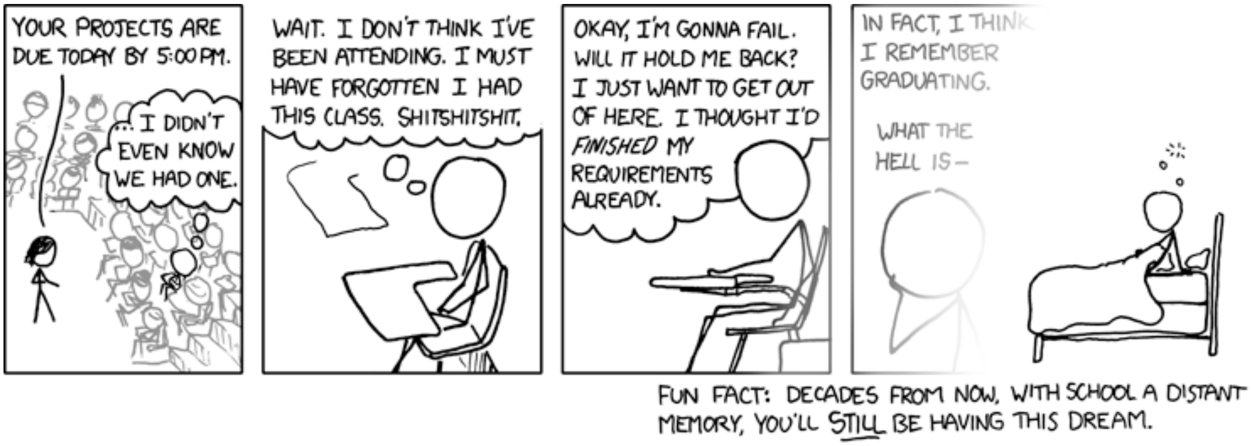
Ian Raphael Sim Ek Yan

00. About


THINGS ON THE FRONT PAGE OF A UNIVERSITY WEBSITE
THINGS PEOPLE GO TO THE SITE LOOKING FOR
CAMPUS PHOTO SLIDESHOW
ALUMNI IN THE NEWS
PROMOTIONS FOR CAMPUS EVENTS
PRESS RELEASES
STATEMENT OF THE SCHOOL'S PHILOSOPHY
LETTER FROM THE PRESIDENT
VIRTUAL TOUR
FULL NAME OF SCHOOL
LIST OF FACULTY PHONE NUMBERS AND EMAILS
CAMPUS ADDRESS
APPLICATION FORMS
ACADEMIC CALENDAR
CAMPUS POLICE PHONE NUMBER
DEPARTMENT/ COURSE LISTS
PARKING INFORMATION
USABLE CAMPUS MAP

Summit Analytics is a web application that empowers educators from the National University of Singapore to make more informed decisions when crafting lessons and assignments. Educators can look forward to various analytical features that will allow them to tailor their module content to their students' needs. The application also allows educators to better track the progress of their students from their course, and also allows administrators to better get an overview of their cohort, based on data visualizations and statistics

01. Product Release Information


YOUR PROJECTS ARE DUE TODAY BY 5:00 PM.
... I DIDN'T EVEN KNOW WE HAD ONE.
WAIT. I DON'T THINK I'VE BEEN ATTENDING. I MUST HAVE FORGOTTEN I HAD THIS CLASS. SHITSHITSHIT.
OKAY, I'M GONNA FAIL. WILL IT HOLD ME BACK? I JUST WANT TO GET OUT OF HERE. I THOUGHT I'D FINISHED MY REQUIREMENTS ALREADY.
IN FACT, I THINK I REMEMBER GRADUATING.
WHAT THE HELL IS—
FUN FACT: DECADES FROM NOW, WITH SCHOOL A DISTANT MEMORY, YOU'LL STILL BE HAVING THIS DREAM.

The product release information section details the requirements we gathered for the Summit application.

01.01. High Level Requirements Elicitation

Requirements for our application were obtained from interviews from the professors and instructors in NUS (Refer to Reference Section 04.01).

Need	Professors need to understand and track student performance and behaviour over the course of the semester and provide prompt intervention when necessary to ensure that course objectives are met.
Justification	Current applications have limited detailed tracking and analysis of student performance,behaviour and course management metrics for educators. In some cases there are no such applications at all. The creation of a new educational analytics application provides opportunities to improve upon the limitations of current systems and promote higher adoption rates of such systems. The needs mentioned above would then be better met.
Scope	To provide clear and powerful insight through analytics on achievements and games.
Major constraint	Allocated time, human resources, changing and lack of clear requirements, and changing data types. This may limit the number of features that we can support from the full list of requirements that we have elicited. A key example would be displaying relevant information to educators and administrators. This can assist professors and department heads in helping them to better achieve course objectives and influence student performance through early intervention. But due to such constraints, we may not be able to finish user views on time.
Major functionality	Track aggregated and individual student performance across assignments using a variety of metrics on time spent on assignments, assignment grades, completion rate and breakdown on specific assignment performance (if relevant). Create intuitive visualisations of the above. Provides some degree of customisation to display only information of interest and achieve ease in making comparisons.
Success factor	Software project, upon completion, must resolve majority of the problems that form the high level need stated in this document ideally by May. This will be determined based on feedback from stakeholders and those from the target audience of users via a holistic satisfaction assessment. The overall aim would be to determine if the key stakeholders agree that our

	solution has helped them to make better sense of the data and that they would potentially use our solution.
User characteristics	Professors, tutors, instructors (Educator View) Department Heads (Administrator View)

01.02. Specifications of Application for Third Iteration

Requirement number	Brief Requirement Description	Status	Details
21	Understand each student's improvement / worsening performance across assignments	Released	From student tab: submission time
23	Categorise students based on the concepts they are weak in	Postponed indefinitely	No way of categorising based on current data
22	Diagnose the difficulty of each assignment based on time metrics and score	Released	From assignments tab
15	Flag out assignments and questions which students face difficulty in	Released	From assignments tab
24	Predict performance (Individual and as a whole course) based on current performance and background	Postponed to next release	Only time based metric available (may not be very reflective of performance)
14a	View aggregated performance of cohort for each assignment (average, standard deviation)	Released	From assignments tab
13a	Zoom into specific errors that students make in each assignment	Postponed Indefinitely	No answer key available to grade solution
19	Track the average time spent on each assignment by the students taking the module	Released	From assignments tab
13	Track and visualise each student's scores across assignments	Released	From students tab, using submission time
25	Prescribe the types of assignments to set and when to release them	Postponed Indefinitely	Assignment type is not correlated with time spent
13b	Identify particularly weak students (in terms of score)	Released	From assignments tab
17	Visualise time gap between the downloading and attending of lectures	Postponed Indefinitely	Data currently unavailable
14	View aggregated performance of cohort:	Rejected(overlap with other 14s)	None
14b	View aggregated performance of cohort For the entire module (average, standard deviation)	Released	From overview tab
20	Track the number of attempts of each assignment	Released	From assignment tab: codecombat data
18	Track the time that individual students spend on assignments (lazy and hardworking)	Released	From students tab
5	Determine the percentage of completed assignments by each student for the course	Released	From assignments tab
3	View the number of courses that have been created in total	Released	Count from selection, admin view

1	Determine the number of students using each educational application	Released	From overview tab: Number of students
2	Determine the number of students who have joined each course	Released	From overview tab: Number of students
16	Visualise time gap between the downloading and submission of assignments	Postponed Indefinitely	No download data time stamp
4	View the number of courses each student is taking	Postponed Indefinitely	No common course ID
6	Detect plagiarism in assignment submissions	Rejected	
7	Enable a chat function between instructors and students	Released	Chat via Telegram bot live
8	Check if students are viewing and responding to messages sent out by instructors	Rejected	
9	Enable students to provide anonymous feedback to instructors regarding the module	Rejected	
10	Automate replies to standard student queries	Rejected	
11	Enable in-class quizzes to measure the understanding of students in real-time during lectures	Rejected	
12	Enable sharing of assignment due-date calendars across modules and instructors	Rejected	
-	User Roles	Released	Educator and Admin View

02. Development

02.01. Process

Our platform is developed using the Agile development process. As can be seen from our schedule we are building our platform in an iterative and incremental manner.

For instance there are two main clear cut iterations, with the second being directly built upon the first in line with the incremental concepts.. A working basic working prototype has already been made available, showing our commitment to the philosophy of working software.

Customer collaboration is another component which we abide by, as seen from our allocated time for check ins and feedback gathering at regular intervals with stakeholders and people from our target audience of educators. Continuous customer interactions allows us to gather proper product requirements and validate them frequently to ensure project success.

Pair programming is also something that we practice in accordance with the individuals and interactions aspect, for instance the overall overview main page is split among two or three people which makes that component suitable for this practice.

Our team strongly believes in the benefits of the Agile model. We believe that it is a realistic approach suitable for frequently changing or uncertain requirements which is the case for this project. Concurrent development allows efficiency given the short time frame. Early development and high customer interaction minimises potential for error and ensures alignment with the intended goals. These factors are why our team chose to follow the Agile manifesto principles as shown above.

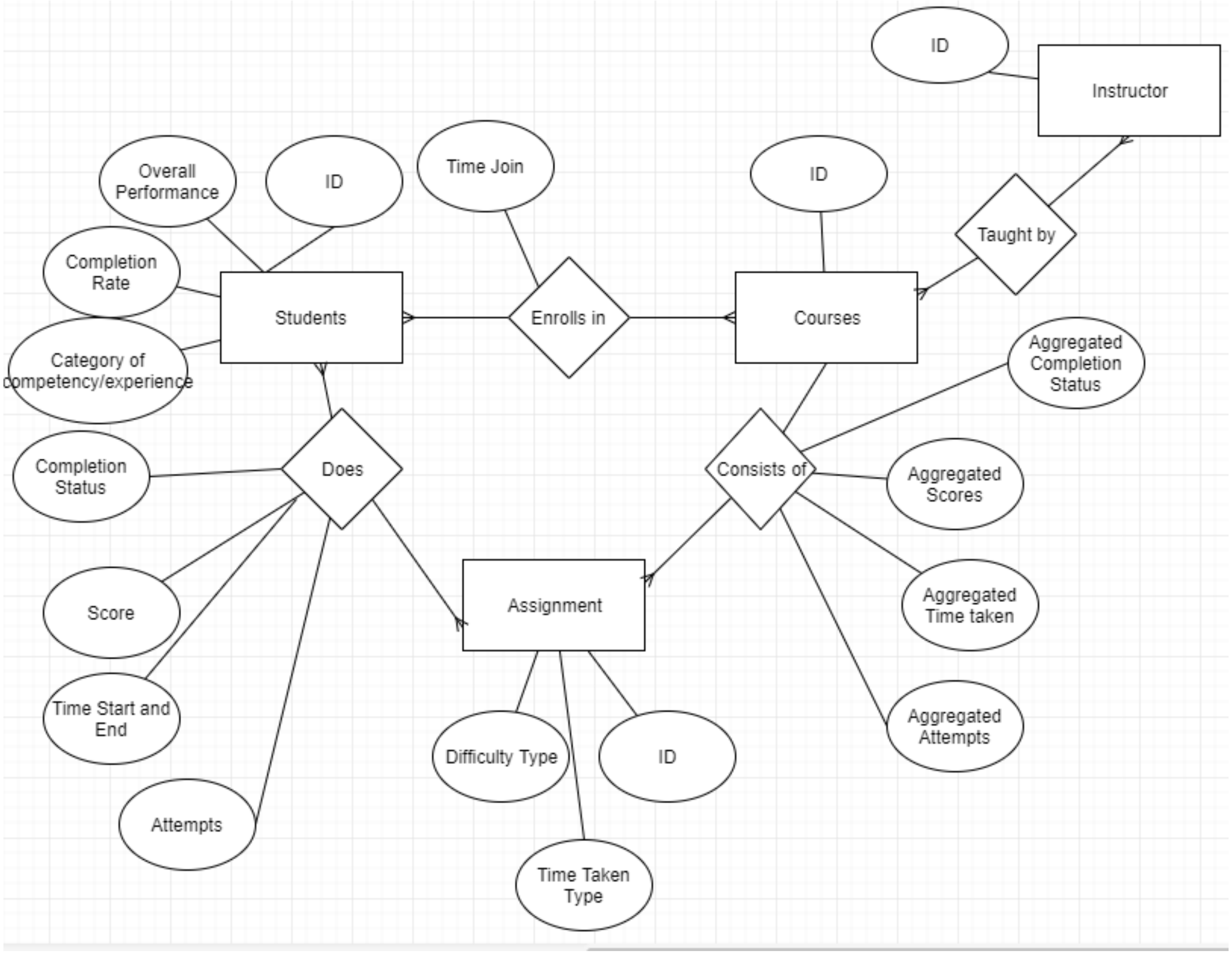
02.02. Architectural and Detailed Design

02.02.01. Database Design

Summit Analytics will be using ReactJS to render information stored on Firebase. To enhance speed of information extraction to generate charts and reduce complexity, no real-time computation will be executed. Instead, the data will be pulled from a frequently updated database where information is pre-calculated to compensate for the weakness of the chosen platform. The structure of this data will be in the form of a json tree.

In total, the analytics required for Summit Analytics falls into 3 broad categories of course management and enrollment, performance and behavior and time related analysis. There will be 5 overarching components to the database which mimic the use cases: (1) course management and enrollment analytics, (2) course performance analytics, (3) overall behaviour and time related analytics for the course, (4) analytics for details of students enrolled in the course and (5)) analytics on course assignments given. At the minimal, each of these components will have its own nested dictionary for easy retrieval of data. We may create more nested dictionaries to further flatten the data structure to reduce how nested the data set is and allow for faster retrieval of data. This is also a best practices for structuring Firebase databases.

The ER Diagram is meant to illustrate roughly how we intend to structure our database. Please refer to section 04.05. for the more complete draft structure of the components to the database. Note that our data in this diagram is not completely normalised to reflect how we intend to store our data eg presence of aggregated attributes.



The technologies that we will be using to develop our app is not dissimilar from what was given as part of the project requirements:

- Rapid web frameworks – React
- Scripting languages: Python, Javascript
- Data schema design - JSON
- Web and mobile interfaces – REST, HTTPS, Firebase, AWS Lambda
- Tracking and analysis of users – Event logs, Jupyter notebooks
- Authentication and authorization – Google login
- Implementing security – Firebase security rules
- Designing and deploying web services – Github, Node

03. Release Notes

Because we often have very frequent releases (every 2 weeks in many cases), this area gives an overview of the entire release history.

03.01. Changelog

03.01.02 28 Feb 2018 - 4 Mar 2018
General:

1. Documentation with formatting
2. Splitting of tasks for individual members

Documentation Changes:

1. Requirement Analysis
2. Task Prioritisation
3. UI Specification (Font, Color)

03.01.02 5 Mar 2018 - 8 Mar 2018
Documentation Changes:

1. Made an actual content page
2. Renamed project title to “Summit Analytics”
3. Re-organised specifications

03.01.03 9 Mar 2018 - 14 Mar 2018
Github Changes:

1. Logout is now functional
2. Sample charts are added to dashboard

General:

1. Achievements Data Dictionary updated
2. Including Codecombat Data in our requirements

03.01.04 15 Mar 2018 - 25 Mar 2018
Python Changes:

1. Created file to download

General:

1. Exploring achievements data
2. Discussion of games and application team merger’s data
3. Playing around with updated achievements dictionary data

Github Changes:

1. Created template for student details
2. Student List implementation

03.01.05 26 Mar 2018 - 31 Mar 2018

Python Changes:

1. Base Firebase file structure developed
2. Assignment Completion Data completed
3. Youtube pause timings data completed
4. Codecombat data achievements incorporated

Github Changes:

1. Redux implemented
2. Course selection dialog implemented
3. Percentile chart added to students page
4. Codecombat chart added to students page

03.01.06 1 Apr 2018 - 6 Apr 2018

Python Changes:

1. Finished codecombat data for completion chart in overview tab
2. Finished students data for overview tab
3. Finished assignments data for overview tab
4. Multiple python scripts are merged into a zip file

Github Changes:

1. Finished Overview Page
2. Added Student Code Combat Charts in student tab

03.01.07 7 Apr 2018 - 12 Apr 2018

Python Changes:

1. Finished summary statistics for codecombat chart

Github Changes:

1. Real Time questions for telegram added
2. Word wrapping for real time telegram questions added
3. Animations to toolbar added
4. Student Codecombat charts added

03.01.08 13 Apr 2018 - 20 Apr 2018

Python Changes:

1. Combined all python file into a lambda update function
2. Created data for both educator and administrator view

Github Changes:

1. Created educator and administrator view
2. Bug fixes
3. Youtube chart added
4. Added authentication

04. Reference Materials

04.01. Raw Interview Responses for Requirement Analysis

04.01.01 Andy Ho (EC1301, Instructor)

1. What analytics tools do you use?

Do not use any

2. What features would you like to see in an analytical software that monitors student performance?

Would like to have functionality that would analyze and understand the students thought process for assignments, as well features that would manage this information, especially easy visualization and intuitive organization.

04.01.02 Assistant Professor Tan Chuan Hoo (NUS Computing DISA Deputy Head)

1. As a Professor, have you used any application that allows you to analyze the performance of your students in detail? (Perhaps in terms of completion status, who took the longest to submit solutions, how many attempts the cohort took on average to complete a particular task, etc.)

I have used the IVLE version for monitoring the students' performance (for submitted assignments and reports) as well as collecting feedbacks through a poll. Usually, the students would submit their written reports in the form of word document file (and pdf version most of the time). They are given the opportunity to resubmit as many times as they like before the deadline. I'll take the last submitted version.

2. If you have, what were the most useful analytical features in your opinion?

Several of them. Including the performance across the assignments. I can see if the students are making good progress and the specific question(s) that most students are not tackling well. I can then discuss those more difficult questions in class. This enables me to adjust my teaching speed and emphasis along the way.

3. What features did you find lacking/missing in the system? Do you feel you might have been able to provide more detailed support and guidance for weaker students if you had these features?

Perhaps, a more dialog-based interaction with the students (incorporated in the original submissions) would be great. This enables me to quickly pinpoint the issues that I like to have a more in-depth discussion with each student. I assume a class size of less than 60. If the class has more than 60 students, then a different approach might be needed.

4. Did you find the user interface of the application easy to use? If you could alter the interface, what would some of these changes be?

There is a learning curve. Specifically, there are many parameters that may not be intuitive. I need to trial and error. I believe this applies to most applications. There is a learning period. On-demand help and explanation of the functions would be great.

5. If you haven't, what analytical features do you feel would have been helpful for the modules you have taught?

I like to have more features that allow me to zoom-in and zoom-out, as well as integrate the assessment based on individual student and cohort, for more encompassing perspective.

6. What benefits do you think these features would have for you and your students? (Perhaps in terms of improving performance, or maybe reducing the amount of time that you spend on repetitive tasks)

I can customize and pace the teaching better. Every intake and cohort of students is different and deserves differentiated care and attention. This is pertinently important as we are not only having students from ONE single degree but students from other programs (e.g., those taking minor, 2nd degree, cross-faculty, exchange) which need to be considered when running courses.

7. Alternatively, what kind of insights would you be speaking from the data of students and their performance? Perhaps any specific metrics you feel is important (which may or may not be covered in existing platforms)?

I would like a more seamless designed app that enables me to compare across assignments (i.e., for a specific student how he/she performs from assignment 1 to assignment X). I like to understand the progress that each student makes. Some of the metrics include the marginal improvement in performance (across assignments), the time gap between downloading the slides and attending the lecture, the time gap between downloading the tutorial questions and the submission of the tutorial answers, etc.

8. What kind of decisions would that help you make? (E.g. finding out suitability of assignments set in terms of difficulty to understand what kind of assignments to set and when to release them?

Yes, what you have provided as examples would be great.

04.01.03 Dr. Kelvin Pang (Tembusu College, Conflict Resolution)

1. As a lecturer, how do you track your student’s progress?

I teach subjects that are more of qualitative in nature, such as dealing with essays, reports etc. For small groups, monitoring performance is easy as it is on a one-to-one basis. For bigger groups, I have strong emphasis on pedagogy & learning outcomes. An analytics platform where students can state their confident level and how certain they are about learning outcomes would be great (such as polls). From there, I would track and make modifications to teaching

2. How do you find existing analytics tools?

New IVLE design is not intuitive, such as where things should be (send email), how to edit folder properties. It is also a poor sharing platform, and I end up turning to Telegram to share with students instead of IVLE. Once you reach level 3000 modules, you don’t really do quizzes. For example, in finance mods, there will be case studies and real-life portfolio analysis.

04.01.04 Associate Professor Irene Woon Mei Yen (IS1105)

1. Have you used any application that allows you to analyze the performance of your students in detail?

Yes, I have used Coursera and IVLE for running my course.

2. What were the most useful analytical features in your opinion?

Most of the feedback are provided based on structured question format e.g. MCQ, MRF. I would have preferred that the system is more intelligent in compiling concepts that students find difficult to grasp. This requires students to type in free form text and system to analyze similarities across answers. For a start, perhaps a word cloud might help?

3. Did you find the user interface of the application easy to use? If you could alter the interface, what would some of these changes be?

They need some getting used to. What helps a lot if quick online responses are available to answer your questions. Coursera's online help is unbeatable. IVLE generally replies within 1/2 days which is on the slow side. A chat would be better.

04.01.05 Dr. Kuan Yee Han (Tembusu College, Biomedical, Conflict Resolution)

1. Have you used any application that allows you to analyse the performance of your students in detail?

I have not used any application of such nature. Only IVLE for managing my classes.

2. What were the most useful analytical features in your opinion?

I would hope to be able to see if my students read the emails for assignment guides, how long did they read the email, have they downloaded the assignment guides. Questions that students have on whether they understand the assignment question

3. Did you find the user interface of the application easy to use? If you could alter the interface, what would some of these changes be?

To minimize email exchanges where clarifications would be required. To see if students have understood the assignments clearly.

04.01.06 Dr. Kelvin Seah Kah Cheng (FASS, Economics, EC1301)

1. As a Professor, have you used any application that allows you to analyze the performance of your students in detail? (Perhaps in terms of completion status, who took the longest to submit solutions, how many attempts the cohort took on average to complete a particular task, etc.)

I've used the IVLE assessment tool to analyze student knowledge (not sure if that qualifies as an app). Typically, there is some bunching of student entries. If you open the assessment for 5 days, most students will complete the assessment either on the first or last day.

2. If you have, what were the most useful analytical features in your opinion? What features did you find lacking/missing in the system? Do you feel you might have been able to provide more detail, support and guidance for weaker students if you had these features?

The system is quite satisfactory. But some of the default settings are not ideal. For instance, the default setting is that students will not be able to resume with the assessment if there is an internet disruption. So, professors will need to remember to change the default. Otherwise, they will get many requests from students to re-do the assessment.

3. Did you find the user interface of the application easy to use? If you could alter the interface, what would some of these changes be?

The system is quite satisfactory. I suppose the app will be useful if it can provide summary statistics on students' overall performance in assessment. So, things like means, standard deviations of assessment scores of each individual student over all tests taken will be useful. It would also be useful to have an app to gauge whether each student is finding the course just right, too difficult, or too easy. At present, we only know these things at the very end, when teaching evaluations are done. It would be helpful to find out what students are thinking as we progress along the course, instead of only at the very end.

04.01.07 Associate Professor Ng Kien Ming (DISEM, IE2110)

1. Yes, I am able to see some of the statistics on student performance.

2. It would be helpful if the system has features that alert me about students facing difficulties, such as those who did not submit homework assignments or who performed consistently below a certain threshold for the continuous assessments. Yes, this might enable me to identify and provide more detailed support and guidance.

3. Yes, which is why the student submissions folders in the Workbin are very helpful for digital assignment submissions.

4. I would be seeking insights on whether certain teaching methods/modes are effective or not, and also whether the teaching pace is appropriate or not.

5. It would help me make the decision on whether to adopt certain teaching methods/modes extensively and also the decision on the teaching pace.

04.02. Analytical Hierarchy Process

AHP	1	2	3	4	5	6	7	8	9	10	11	12	13	13a	13b	14	14a
1	1	1	0.5	0.5	3								4	4	3	4	4
2		1	1	0.5	2								4	4	4	3	3
3			1	2	3								3	2	3	3	2
4				1	2								3	4	3	4	3
5					1								1	2	2	4	3
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13													1	1	1	1	1
13a														1	2	0.5	0.5
13b															1	1	1
14																1	1
14a																	1
14b																	
15																	
16																	
17																	
18																	
19																	

20																	
21																	
22																	
23																	
24																	
25																	

AHP (Continued)

AHP	14b	15	16	17	18	19	20	21	22	23	24	25	Source
1	3	4	1	0.5	2	1	3	3	3	4	2	4	Prof Boesch
2	3	5	2	1	1	1	2	4	3	3	2	2	Prof Boesch
3	2	3	0.5	0.5	3	3	2	0.5	3	2	4	0.5	Prof Boesch
4	3	4	1	0.5	2	2	3	3	3	3	3	2	Prof Boesch
5	3	2	1	1	1	2	1	3	3	3	2	2	Prof Boesch
6													
7													
8													
9													
10													
11													
12													
13	1	2	0.25	0.25	0.33	0.33	0.33	1	2	1	1	0.5	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah
13a	0.5	1	0.25	0.33	0.33	0.33	1	2	0.5	3	1	0.5	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah
13b	1	2	0.25	0.25	0.5	0.5	0.5	2	1	3	1	0.5	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah
14	1	2	0.5	0.5	1	1	1	3	3	2	4	3	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah

14a	1	1	0.5	0.5	2	1	2	4	4	3	4	4	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah
14b	1	1	0.5	0.5	2	1	2	4	4	3	4	4	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah
15		1	0.5	0.5	1	0.75	1	4	3	2	3	3	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah/Prof Yee Han
16			1	1	2	2	2	4	4	3	4	3	Prof Boesch/Prof Kelvin Seah/ Prof Yee Han
17				1	2	2	2	4	4	3	4	3	Prof Boesch/Prof Kelvin Seah/ Prof Yee Han
18					1	1	1	4	4	3	4	3	Prof Boesch/Prof Kelvin Seah/ Prof Yee Han
19						1	1	0.5	3	0.5	1	0.5	Prof Boesch/Prof Kelvin Seah/ Prof Yee Han
20							1	4	4	3	4	1	Prof Boesch/Prof Kelvin Seah/ Prof Yee Han
21								1	0.5	1	0.5	0.5	Prof Kelvin Pang/ Prof Irene Woon/ Prof Boesch/Andy Teo
22									1	4	1	1	Prof Kelvin Pang/ Prof Irene Woon/ Prof Boesch
23										1	0.5	0.5	Prof Kelvin Pang/ Prof Irene Woon/Prof Yee Han
24											1	1	Prof Boesch
25												1	Prof Boesch

The results are shown below:

Requirement Number	Requirement Value %	Priority Level	Requirement Status	Pre-requisite Requirements
21	7.86%	1	Postponed to next release	18, 19
23	7.62%	1	Postponed indefinitely	Data currently unavailable
22	7.18%	1	Postponed to next release	14a, 19
15	6.73%	1	Accepted for this release	None
24	6.72%	1	Postponed to next release	13, 18
14a	5.88%	2	Accepted for this release	None
13a	5.67%	2	Postponed to next release	Data currently unavailable
19	5.66%	2	Accepted for this release	None
13	5.56%	2	Accepted for this release	None
25	5.41%	2	Postponed to next release	14a, 19
13b	5.26%	3	Accepted for this release	None
17	5.03%	3	Postponed indefinitely	Data currently unavailable
14	3.94%	3	Accepted for this release	None
14b	3.56%	3	Accepted for this release	None
20	3.35%	3	Accepted for this release	None
18	3.02%	4	Accepted for this release	None
5	2.59%	4	Postponed to next release	None
3	2.26%	4	Postponed to next release	None
1	1.78%	4	Postponed indefinitely	Data currently unavailable
2	1.68%	5	Postponed to next release	None
16	1.64%	5	Postponed indefinitely	Data currently unavailable
4	1.61%	5	Postponed indefinitely	Data currently unavailable
6	0	NA	Rejected	
7	0	NA	Rejected	
8	0	NA	Rejected	
9	0	NA	Rejected	
10	0	NA	Rejected	
11	0	NA	Rejected	
12	0	NA	Rejected	

04.03. Iteration of Requirements

04.03.01. First Iteration

Requirement number	Brief Requirement Description	Requirement Source	Percent of requirement value	Priority level	Requirement status	Pre-requisite requirements
21	Understand the cause of each student's improvement / worsening performance across assignments	Prof Kelvin Pang/ Prof Irene Woon/ Prof Boesch/Andy Teo	7.86%	1	Postponed to next release	18, 19
23	Categorise students based on the concepts they are weak in	Prof Kelvin Pang/ Prof Irene Woon/Prof Yee Han	7.62%	1	Postponed indefinitely	Data currently unavailable
22	Diagnose the difficulty of each assignment based on time metrics and score	Prof Kelvin Pang/ Prof Irene Woon/ Prof Boesch	7.18%	1	Postponed to next release	14a, 19
15	Flag out assignments and questions which students face difficulty in	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah/Prof Yee Han	6.73%	1	Accepted for this release	None
24	Predict performance (Individual and as a whole course) based on current performance and background	Prof Boesch	6.72%	1	Postponed to next release	13, 18
14a	View aggregated performance of cohort for each assignment (average, standard deviation)	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah	5.88%	2	Accepted for this release	None
13a	Zoom into specific errors that students make in each assignment	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah	5.67%	2	Postponed to next release	Data currently unavailable
19	Track the average time spent on each assignment by the students taking the module	Prof Boesch/Prof Kelvin Seah/ Prof Yee Han	5.66%	2	Accepted for this release	None
13	Track and visualise each student's scores across assignments	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah	5.56%	2	Accepted for this release	None

25	Prescribe the types of assignments to set and when to release them	Prof Boesch	5.41%	2	Postponed to next release	14a, 19
13b	Identify particularly weak students (in terms of score)	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah	5.26%	3	Accepted for this release	None
17	Visualise time gap between the downloading and attending of lectures	Prof Boesch/Prof Kelvin Seah/ Prof Yee Han	5.03%	3	Postponed indefinitely	Data currently unavailable
14	View aggregated performance of cohort:	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah	3.94%	3	Rejected(due to overlap with other 14s)	None
14b	View aggregated performance of cohort For the entire module (average, standard deviation)	Prof Chuan Hoon/ Prof Boesch/Prof Kelvin Seah	3.56%	3	Accepted for this release	None
20	Track the number of attempts of each assignment	Prof Boesch	3.35%	3	Accepted for this release	None
18	Track the time that individual students spend on assignments (lazy and hardworking)	Prof Boesch	3.02%	4	Accepted for this release	None
5	Determine the percentage of completed assignments by each student for the course	Prof Boesch	2.59%	4	Postponed to next release	None
3	View the number of courses that have been created in total	Prof Boesch	2.26%	4	Postponed to next release	None
1	Determine the number of students using each educational application	Prof Boesch	1.78%	4	Postponed indefinitely	Data currently unavailable
2	Determine the number of students who have joined each course	Prof Boesch	1.68%	5	Postponed to next release	None
16	Visualise time gap between the downloading and submission of assignments	Prof Boesch/Prof Kelvin Seah	1.64%	5	Postponed indefinitely	Data currently unavailable
4	View the number of courses each student is taking	Prof Boesch	1.61%	5	Postponed indefinitely	Data currently unavailable

6	Detect plagiarism in assignment submissions	Prof Boesch	0	NA	Rejected	
7	Enable a chat function between instructors and students	Prof Kelvin Pang	0	NA	Rejected	
8	Check if students are viewing and responding to messages sent out by instructors	Prof Yee Han	0	NA	Rejected	
9	Enable students to provide anonymous feedback to instructors regarding the module	Prof Kelvin Pang	0	NA	Rejected	
10	Automate replies to standard student queries	Prof Irene Woon	0	NA	Rejected	
11	Enable in-class quizzes to measure the understanding of students in real-time during lectures	Prof Irene Woon	0	NA	Rejected	
12	Enable sharing of assignment due-date calendars across modules and instructors	Prof Boesch	0	NA	Rejected	

04.03.02. Second Iteration

Requirement number	Brief Requirement Description	Status	Details
21	Understand each student’s improvement / worsening performance across assignments	Released	From student tab: submission time
23	Categorise students based on the concepts they are weak in	Postponed indefinitely	No way of categorising based on current data
22	Diagnose the difficulty of each assignment based on time metrics and score	Released	From assignments tab
15	Flag out assignments and questions which students face difficulty in	Released	From assignments tab
24	Predict performance (Individual and as a whole course) based on current performance and background	Postponed to next release	Only time based metric available (may not be very reflective of performance)
14a	View aggregated performance of cohort for each assignment (average, standard deviation)	Released	From assignments tab
13a	Zoom into specific errors that students make in each assignment	Postponed Indefinitely	No answer key available to grade solution
19	Track the average time spent on each assignment by the students taking the module	Released	From assignments tab
13	Track and visualise each student’s scores across assignments	Released	From students tab, using submission time
25	Prescribe the types of assignments to set and when to release them	Postponed Indefinitely	Assignment type is not correlated with time spent
13b	Identify particularly weak students (in terms of score)	Released	From assignments tab
17	Visualise time gap between the downloading and attending of lectures	Postponed Indefinitely	Data currently unavailable
14	View aggregated performance of cohort:	Rejected(overlap with other 14s)	None
14b	View aggregated performance of cohort For the entire module (average, standard deviation)	Released	From overview tab
20	Track the number of attempts of each assignment	Released	From assignment tab: codecombat data
18	Track the time that individual students spend on assignments (lazy and hardworking)	Released	From students tab
5	Determine the percentage of completed assignments by each student for the course	Released	From assignments tab
3	View the number of courses that have been created in total	Released	Count from selection, admin view
1	Determine the number of students using each educational application	Released	From overview tab:

			Number of students
2	Determine the number of students who have joined each course	Released	From overview tab: Number of students
16	Visualise time gap between the downloading and submission of assignments	Postponed Indefinitely	No download data time stamp
4	View the number of courses each student is taking	Postponed Indefinitely	No common course ID
6	Detect plagiarism in assignment submissions	Rejected	
7	Enable a chat function between instructors and students	Released	Chat via Telegram bot live
8	Check if students are viewing and responding to messages sent out by instructors	Rejected	
9	Enable students to provide anonymous feedback to instructors regarding the module	Rejected	
10	Automate replies to standard student queries	Rejected	
11	Enable in-class quizzes to measure the understanding of students in real-time during lectures	Rejected	
12	Enable sharing of assignment due-date calendars across modules and instructors	Rejected	

04.04. Achievements Data Dictionary

From Prof Boesch’s Document

API Options
["users", "cohorts", "courses","assignments","solutions","courseMembers","cohortCourses", "usage"]

Your team will receive its API token via email. Remember to keep this token secret from others. There is a limit to how much data you can download with your token and all downloads are tracked.

There is also an assignment in Achievements with a notebook problem that demonstrates how to download data to a Colaboratory notebook environment using your API key.

API URLs
https://us-central1-achievements-prod.cloudfunctions.net/api?token={TOKEN}&data={option}
https://us-central1-achievements-prod.cloudfunctions.net/api?token=So_Fancy&data=cohorts
https://us-central1-achievements-prod.cloudfunctions.net/api?token=So_Fancy&data=courses

API Data

- assignments
    - assignments/{$courseKey}/{$assignmentKey}
        - count
        - deadline
        - details
        - level
        - name
        - open
        - orderIndex
        - questionType
        - solutionVisible
        - visible
- cohortCourses
    - cohortCourses/{$cohortKey}/{$courseKey}
        - name
        - participants
        - progress
- cohorts
    - cohorts/{$cohortKey}
        - description
        - instructorName

- isPublic
- name
- owner

- courseMembers
    - courseMembers/{$courseKey}/{$uid} -> true
- courses
    - courses/{$courseKey}
        - instructorName
        - isPublic
        - name
        - owner
- solutions
    - solutions/{$courseKey}/{$uid}/{$solutionKey}
        - createdAt
        - value
- userAchievements
    - userAchievements/{$uid}/CodeCombat
        - achievements:
            - Dungeons-of-kithgard
                - complete: true
                - name: "Dungeons of Kithgard"
        - id: "artificial-intelligence"
        - lastUpdate: 1520578759259
        - totalAchievements: 190
- users
    - users/{$uid}
        - acceptedEULA
        - displayName
        - photoURL

Achievements Event Types:

Following are the events types logged in the achievements database:

1. LOGIN_MENU_OPEN - User opened the upper right login menu
2. LOGIN_MENU_CLOSE - User closed upper right login menu
3. UPDATE_ACHIEVEMENTS_DATA - backend process added a new Code Combat level completion for a user.
4. COURSE_ASSIGNMENTS_OPEN - user viewed course and opened assignments tab
5. COURSE_MEMBERS_FETCH_SUCCESS - members in course fetched

6. COURSE_MEMBER_ACHIEVEMENTS_REFETCH by Prof Boesch - members Code Combat achievements fetched12:21:24
7. ASSIGNMENT_SOLUTION_REQUEST - User submitted a solution for an assignment
8. ASSIGNMENT_SOLUTION_SUCCESS - Successful assignment submission
9. NOTIFICATION_SHOW - User shown notification in bottom left corner
10. NOTIFICATION_HIDE - User notification message hidden

When someone hits log-in:

1. COURSE_JOINED_FETCH_SUCCESS - Upon logging in, a request is sent to fetch the courses joined.
2. SIGN_IN_SUCCESS - Logged upon successful sign in.

When on the Profile page:

1. DISPLAY_NAME_EDIT_TOGGLE - Upon clicking on edit button to edit Display name.
2. DISPLAY_NAME_UPDATE_SUCCESS - Self explanatory.
3. DISPLAY_NAME_EDIT_TOGGLE - When the logged in user clicks on edit button next to the display-name.
4. EXTERNAL_PROFILE_DIALOG_SHOW - Upon clicking on “Add-profile”.
5. EXTERNAL_PROFILE_UPDATE_REQUEST - Upon clicking “commit” to add profile, user’s achievements fetch request is sent.
6. EXTERNAL_PROFILE_REFRESH_REQUEST -
7. EXTERNAL_PROFILE_REMOVE_DIALOG_SHOW - Request sent upon clicking “Remove Profile”.
8. EXTERNAL_PROFILE_REMOVE_REQUEST - Self explanatory.
9. EXTERNAL_PROFILE_REMOVE_SUCCESS - Self explanatory.
10. EXTERNAL_PROFILE_REFRESH_REQUEST - Profile refresh request sent in by user.

When on the ‘Courses’ tab:

1. ROUTE_CHANGE - Upon a route change.
2. MAIN_DRAWER_TOGGLE - When trying to toggle to another/same tab from the Side-drawer.
3. COURSE_SHOW_NEW_DIALOG - Upon clicking on “Add new Course” from Courses tab.
4. COURSE_HIDE_DIALOG - Self explanatory.
5. COURSE_NEW_SUCCESS - Upon successfully adding a new course.
6. COURSE_SWITCH_TAB - When switching between the 3 tabs inside Courses.

When inside a Course:

1. ASSIGNMENT_SUBMIT_REQUEST - Upon clicking on “Submit
2. EXTERNAL_PROFILE_UPDATE_REQUEST - Whenever there is a submission to “Enter Code Combat Profile” question, an external update request is sent to Codecombat to fetch their badges.
3. ASSIGNMENT_SOLUTION_REQUEST
4. ASSIGNMENT_SOLUTION_SUCCESS - Upon successful submission of the assignment.

5. EXTERNAL_PROFILE_DIALOG_HIDE - Upon clicking on “cancel” on pop-up for submission of Code combat name.

(These will be for an instructor of a course only. Students don’t see multiple tabs.)

1. ASSIGNMENT_SWITCH_TAB -
2. ASSIGNMENTS_EDITOR_TABLE_SHOWN - Upon toggling to ‘Edit’ tab.
3. ASSIGNMENT_SHOW_ADD_DIALOG - Upon clicking “Add assignment” on Edit page.
4. ASSIGNMENT_ADD_REQUEST -
5. ASSIGNMENT_ADD_SUCCESS -
6. ASSIGNMENT_REFRESH_PROFILES_REQUEST - Request received to refresh the achievements table.
7. ASSIGNMENT_REFRESH_PROFILES_SUCCESS - Request successful to refresh the achievements table.
8. ASSIGNMENT_QUICK_UPDATE_REQUEST - When the course instructor submits an edit request for an assignment.
9. ASSIGNMENT_QUICK_UPDATE_SUCCESS
10. COURSE_REMOVE_STUDENT_REQUEST - When an instructor removes a student from the course.
11. COURSE_REMOVE_STUDENT_SUCCESS

Alphabetical list of all event types:

ACCEPT_EULA_REQUEST
ACCEPT_EULA_SUCCESS
ADD_COHORT_DIALOG_HIDE
ADD_COHORT_DIALOG_SHOW
ADD_COHORT_FAIL
ADD_COHORT_REQUEST
ADD_COHORT_SUCCESS
ASSIGNMENTS_ASSISTANTS_DIALOG_SHOW
ASSIGNMENTS_ASSISTANTS_SHOW_REQUEST
ASSIGNMENTS_EDITOR_TABLE_SHOWN
ASSIGNMENT_ADD_ASSISTANT_REQUEST
ASSIGNMENT_ADD_ASSISTANT_SUCCESS
ASSIGNMENT_ADD_REQUEST
ASSIGNMENT_ADD_SUCCESS
ASSIGNMENT_ASSISTANT_FOUND
ASSIGNMENT_ASSISTANT_KEY_CHANGE
ASSIGNMENT_CLOSE_DIALOG
ASSIGNMENT_QUICK_UPDATE_REQUEST
ASSIGNMENT_QUICK_UPDATE_SUCCESS
ASSIGNMENT_REFRESH_PROFILES_FAIL
ASSIGNMENT_REFRESH_PROFILES_REQUEST
ASSIGNMENT_REFRESH_PROFILES_SUCCESS
ASSIGNMENT_REORDER_REQUEST

ASSIGNMENT_REORDER_SUCCESS
ASSIGNMENT_SHOW_ADD_DIALOG
ASSIGNMENT_SHOW_EDIT_DIALOG
ASSIGNMENT_SOLUTION_FAIL
ASSIGNMENT_SOLUTION_REQUEST
ASSIGNMENT_SOLUTION_SUCCESS
ASSIGNMENT_SUBMIT_REQUEST
ASSIGNMENT_SWITCH_TAB
COHORTS_CHANGE_TAB
COHORT_COURSES_RECALCULATE_REQUEST
COURSE_ASSIGNMENTS_CLOSE
COURSE_ASSIGNMENTS_OPEN
COURSE_HIDE_DIALOG
COURSE_JOINED_FETCH_SUCCESS
COURSE_MEMBERS_FETCH_SUCCESS
COURSE_MEMBER_ACHIEVEMENTS_REFETCH
COURSE_NEW_FAIL
COURSE_NEW_SUCCESS
COURSE_PASSWORD_ENTER_FAIL
COURSE_PASSWORD_ENTER_SUCCESS
COURSE_REMOVE_DIALOG_SHOW
COURSE_REMOVE_FAIL
COURSE_REMOVE_REQUEST
COURSE_REMOVE_STUDENT_DIALOG_SHOW
COURSE_REMOVE_STUDENT_FAIL
COURSE_REMOVE_STUDENT_REQUEST
COURSE_REMOVE_STUDENT_SUCCESS
COURSE_REMOVE_SUCCESS
COURSE_SHOW_NEW_DIALOG
COURSE_SWITCH_TAB
DISPLAY_NAME_EDIT_TOGGLE
DISPLAY_NAME_UPDATE_REQUEST
DISPLAY_NAME_UPDATE_SUCCESS
EXTERNAL_PROFILE_DIALOG_HIDE
EXTERNAL_PROFILE_DIALOG_SHOW
EXTERNAL_PROFILE_REFRESH_FAIL
EXTERNAL_PROFILE_REFRESH_REQUEST
EXTERNAL_PROFILE_REFRESH_SUCCESS
EXTERNAL_PROFILE_REMOVE_DIALOG_HIDE
EXTERNAL_PROFILE_REMOVE_DIALOG_SHOW
EXTERNAL_PROFILE_REMOVE_REQUEST
EXTERNAL_PROFILE_REMOVE_SUCCESS
EXTERNAL_PROFILE_UPDATE_FAIL
EXTERNAL_PROFILE_UPDATE_REQUEST

EXTERNAL_PROFILE_UPDATE_SUCCESS
LOGIN_MENU_CLOSE
LOGIN_MENU_OPEN
MAIN_DRAWER_TOGGLE
NOTIFICATION_HIDE
NOTIFICATION_SHOW
PATH_CHANGE_REQUEST
PATH_CHANGE_SUCCESS
PATH_DIALOG_HIDE
PATH_DIALOG_SHOW
PATH_PROBLEM_DIALOG_SHOW
PATH_SELECT
ROUTE_CHANGE
SHOW_ACCEPT_EULA_DIALOG
SIGN_IN_FAIL
SIGN_IN_REQUEST
SIGN_IN_SUCCESS
SIGN_OUT_REQUEST
UPDATE_ACHIEVEMENTS_DATA

04.05. Database Schema Design

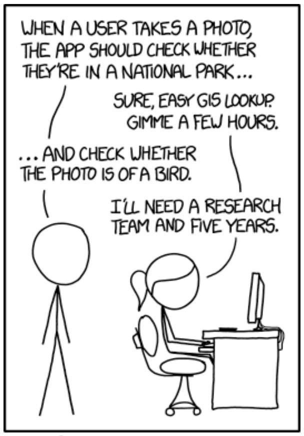
Variable string names are denoted by “” and hold no actual value.

```
 //This includes information for course and student management under management and
enrolment

{“manage_enrol_course”: {
  “num_courses”: integer,
  “courses”: {course_id: {       //list of courses
                          “Instructor”: [userID (string), displayName(string)],
                          “students”: {studentID: {
                                  “details”: [displayname, [joinedDate (timestamp),
                                  joinedDateFromToday (integer)],
                                  “experienced”: Boolean,
                                  listCourses: {courseID: courseName (string)}, “numCourses”:
                                  integer}}

                          “assignmentTypesDetails”:{
                                  typeA: [medianCompletionRate (percentage),
                                  totalSubmittedSolutions (integer), medianScores (integer) ]},
                          “overallAssignments”: [completionRate (percentage), score (integer)],

                          “joinedUsers”:{
                                  timestamp: {
                                          “numDays”: integer,
                                          “numStudents”: integer, “enrolmentGrowthPercentage”:
                                          integer}}
                          “PercentageNewStudents”: integer,
                  }}
}}
```

```
{“performance_Behaviour_student”: {course_id: { //list of course ids
    “student_list”: {studentID: {
                "displayName": name (string),
                “assignmentDetails”: {assignmentID: {
                    “assignmentType”: string,
                    “submissionDate”: timestamp (optional),
                    “submissionPromptness”: ((deadline-
                    submissionDate)/(deadline-addedDate))       //in
                    percentage (optional)
                    “completed”: Boolean,
                    “attempted”: Boolean,
                    “numAttempts”: integer,
                    “timeComplete”: [timestamp, location (home or school)],
                    “timeTaken”: integer,
                    “score”: integer,
                    predictedScore: integer (optional)     }}, //prediction is
                    only for code combat data
                “assignmentAggregate”:{
                    typeA: {
                        “completion”: [medianCompletionRate
                        (percentage), totalSubmittedSolutions (integer),
                        averageAttempts (integer),
                        medianSubmissionPromptness (percentage)],
                        “scores”: [averageScore, 25thPercentile,
                        medianScore, 75thpercentile,
                        standardDeviationScore, lowestScore,
                        maximumScore],
                        “time”: [homeAttemptRates, schoolAttemptRates,
                        averageTime, 25thPercentile, medianTime,
                        75thpercentile, standardDeviationTime, fastest,
                        slowest, totalTime],
                        “performanceReview”: {
                            overallPercentileRanking: integer,
                            “performanceOverview”: {1:integer} }},
                            “top5Assignments”:[assignmentID],
                            “bottom5Assignments”:[assignmentID]}
                        }},
        “Completion”: [assignmentCompletionRate (percentage),
        averageAttempts (integer), totalAttempts (integer),
        mainSchoolCompletion (boolean),
        medianSubmissionPromptness (percentage)]
        “aggregateOwnScores”: [averageScore, 25thPercentile,
        medianScore, 75thpercentile, standardDeviationScore,
        lowestScore, maximumScore],
        “aggregateOwnTiming”: [homeAttemptRates,
        schoolAttemptRates, averageTime, 25thPercentile, medianTime,
        75thpercentile, standardDeviationTime, fastest, slowest,
        totalTime],

overallPercentileRanking: integer,
“performanceOverview”: {1: integer} //assignment&Score
        “topAssignmentTypes”: string,
“weakAssignmentTypes”: string
}}}}}
```

https://docs.google.com/document/d/1eYTx_fRAyPMsgCzrbX6lMS_j9M_jagdfiv_caoXC9JA/edit

04.06. Elicited Detailed Requirements, Requirement Analysis and Clustering by Viewpoint Oriented Requirements Definition


WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
...AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.

IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

Based on the high level requirements stated above, we conducted rigorous interviews with professors and instructors from the National University of Singapore to validate the requirements. From these interviews, we have obtained a list of detailed requirements that are most relevant to the high level needs mentioned above and are applicable to the current and stated future functionality of the achievements and games applications.

The requirements below are mainly catered to educators, thus, functionalities which apply to students will be marked with an "*" asterisk.

Course is defined as the name of the institution students belong to and cohort is defined as the competition level based on the current form of the achievements app..

We considered the design related issues and hence proposed a new UI that has the following characteristics:
i. Intuitive Organisation
ii. Avoid steep learning curve as seen in IVLE (Such as by including Frequently Asked Questions)
iii. Options to customise look (via settings)

04.06.01. Non-functional requirements

i. Login page to ensure that only instructors have access to course and cohort level analytics

ii. Interface requirements:

a. Landing page that provides an overview for instructors (consisting of the most critical analytics and charts)

b. Students page that allows an instructor to scroll through the list of students in his course(s), and which also allows him/her to select a student to find out more about the student in detail

c. A student details page that contains relevant performance and behaviour information for a particular student

d. Assignments page that allows an instructor to scroll through the list of students in his course(s), and which also allows him/her to select a student to find out more about the student in detail.

e. Being able to easily display only selected relevant information such as particular students,assignments, or courses.

f. Being able to easily search for the relevant courses/students/assignments

04.06.02. Functional requirements

These requirements are applicable to all types of assignments in the achievements application, such as codecombat, jupyter notebooks and youtube videos. There will thus be a minimal amount of adaptation necessary for newly released features from the Achievements team.

04.06.02.01 Management and enrollment

i. Course related analytics:

a. Descriptive and Diagnostic:

1. Total number of students
2. Determine the number of students who have joined each course and show the growth (overall and per day)
3. Indicate how long have the users joined on average (distinguish between experienced and new based on when you joined)
4. What percentage of students are new students

b. Predictive and Prescriptive:

1. Predict number of users who will join and from where

ii.. Student related analytics:

a. View the number of courses each student is taking

b. View if the student is a new student

c. View how long the student has joined

04.06.02.02 Performance

i. Individual Students:

a. Descriptive and Diagnostic:
   i. Determine the percentage of completed assignments by each student for the course
   ii. Track and visualise each student's performance (ie. number of assignments completed, promptness of completing assignments)
   iii. For each assignment (completion rate and score)
   iv. Understand the cause of each student's improvement / worsening performance across assignments (diagnostic)

b. Predictive and Prescriptive:
   i. Predict future performance based on current performance and background

ii. Individual Assignments:

a. Descriptive and Diagnostic:
   i. Flag out if that particular assignments/ there are questions which students face difficulty in
   ii. Display statistics for each assignment (completion rate and score, time taken etc)
   iii. Diagnose the difficulty of each assignment based on time metrics and completion rate
   iv. Display how many total solutions have been submitted per assignment

b. Predictive and Prescriptive:
   i. Predict an estimated time of completion or difficulty category (eg short or long, easy or hard) for the assignment

iii. Overall Course :

a. Descriptive and Diagnostic:
   i. What is the mean/mode/median number of assignments completed for each course?
   ii. Determine the percentage of completed assignments by each student for the course
   iii. Understanding causes of differences in performance between students (Display assignments where many people struggled/got wrong/have not done)
   iv. Track performance of students to flag out students with declining performance(and improving)
   v. Flag out key differences in courses
   vi. Categorise students based on the assignments they are weak in
   vii. Highlight particularly weak students (in terms of completion and score rate) for each course to help
   viii. Identify top students for each course (in terms of completion and score rate)

b. Predictive and Prescriptive:
   i. Prescribe the types of assignments to set and when to release them

04.06.02.03 Behaviour/Time

i. Individual Assignments:

a. Length of time taken to complete each assignment
b. Average number of attempts for each assignment

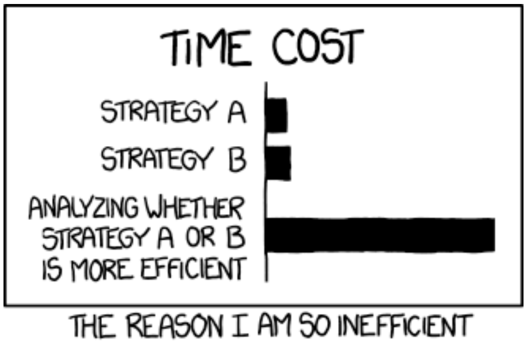
ii. Individual Students:

a. Total amount of time spent on assignments
b. Total number of attempts there
c. Whether student largely completes assignment at home or at school

iii. Overall Course and Cohort:

a. Total amount of time spent by each course and cohort for assignments on average
b. Total number of attempts made by students for assignments in each course on average

04.07. Project Timeline (Old)


TIME COST
STRATEGY A
STRATEGY B
ANALYZING WHETHER STRATEGY A OR B IS MORE EFFICIENT
THE REASON I AM SO INEFFICIENT

04.07.01. Prioritisation

While developing this document, the team went through a requirements prioritisation exercise two times. The first time, we derived requirements from discussion with various stakeholders and held an internal review to determine if they were relevant and achievable. The list of requirements that we consolidated formed the basis for further discussions (Refer to Reference Section 04.03).

As we learnt more about the requirements, we revised the requirements prioritisation to arrive at the current version below. We intend to have two major iterations, and thus split our requirements into two. The first iteration contains the essential functional features in our application, while the second iteration addresses non-functional and relatively advanced features that require certain "prerequisites" from iteration 1.

Both prioritisation exercises made use of the Analytical Hierarchy Process (Refer to Reference Section 04.02).

Iteration	Requirement(s) Number	Details	Requirement Source
1	Functional 1 (i)(a),1(ii)	Management and enrollment Descriptive and Diagnostic Analytics	Prof Boesch, Prof Chuan Hoo, Prof Yee Han, Prof Kelvin Pang
1	Functional 2(iii)(a)	Performance of Overall Course Descriptive and Diagnostic analytics	Prof Boesch, Prof Kelvin Seah, Prof Chuan Hoo
1	Functional 3(iii)	Behaviour/Time metrics of Overall Course Descriptive and diagnostic analytics	Prof Boesch, Prof Yee Han, Prof Kelvin Pang
1	Functional 2(ii)(a), 3(i)	Performance of Individual Assignments	Prof Boesch, Prof Irene Woon,Prof

		Descriptive and Diagnostic analytics	Kelvin Seah
1	Functional 2(i)(a), 3(ii)	Performance of Individual Students Descriptive and Diagnostic analytics	Prof Boesch, Prof Kien Ming, Prof Kelvin Seah, Prof Chuan Hoo
2	Non Functional 1	Homepage Login, individual user account information	Prof Boesch
2	Non Functional 2(e)	Display only selected relevant students, assignments, or courses	Prof Boesch, Prof Chuan Hoo
2	Non Functional 2(f)	Course/cohort search functionality	Prof Boesch, Prof Chuan Hoo
2	Functional 1(i)(b), 2(i)(b)	Student Performance and enrollment predictive analytics	Prof Boesch, Prof Kien Ming
2	Functional 2(ii)(b), 2(iii)(b)	Predictive and Prescriptive Analytics to improve assignment setting	Prof Boesch, Prof Chuan Hoo

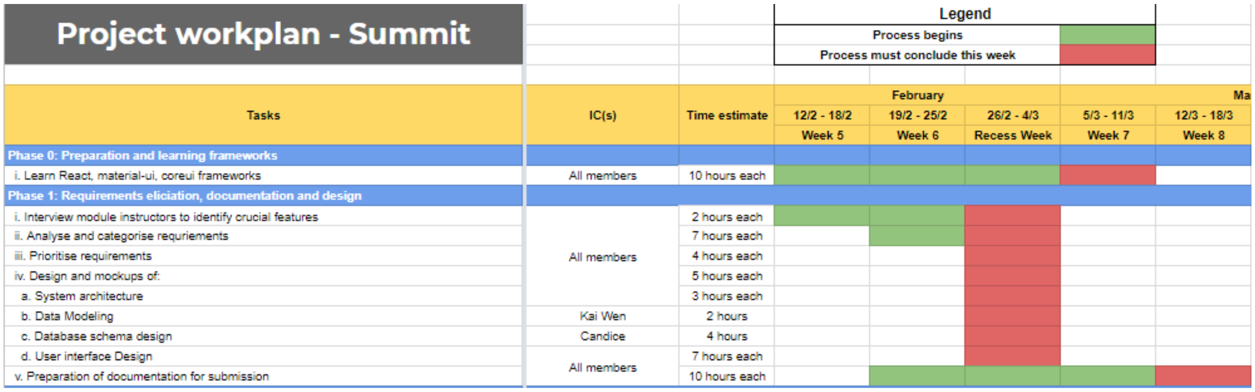
04.07.02. Project Work Plan

To facilitate the division of labour and the enable the above tasks to be completed on time, the team worked together to develop the following work plan, which details all the tasks that need to be completed by the release date of the application (week 13).

Do to limitations in document width, the image seen might be difficult to read. The full workplan can be accessed at the link below (Please feel free to contact the team if the document sharing is broken). In addition, do note that the schedule serves mainly as a general guide to the major milestones, and that secondary tasks related to the assigned job scopes may not be listed in the work plan e.g. continuous communication of updates and reviews of each other's work.

https://docs.google.com/spreadsheets/d/1tg9huheBFNa9HKNtgn6lB3MP0Qz9N1OCBuOmSfpUPPA/edit#gid=0

Phase 0 and 1: Preparation,requirements elicitation, documentation and design

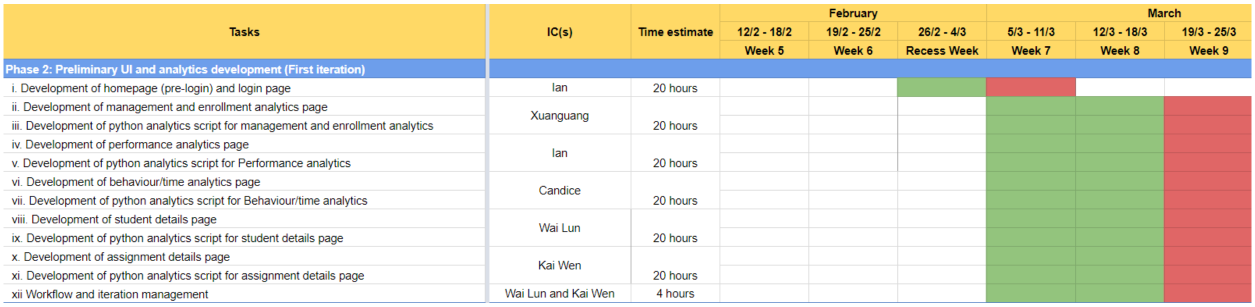


Project workplan - Summit

Legend	
Process begins	🟩
Process must conclude this week	🟥

Tasks	IC(s)	Time estimate	February			Ma	
			12/2 - 18/2	19/2 - 25/2	26/2 - 4/3	5/3 - 11/3	12/3 - 18/3
			Week 5	Week 6	Recess Week	Week 7	Week 8
Phase 0: Preparation and learning frameworks							
i. Learn React, material-ui, coreui frameworks	All members	10 hours each	🟩	🟩	🟩	🟥	
Phase 1: Requirements eliciation, documentation and design							
i. Interview module instructors to identify crucial features	All members	2 hours each	🟩	🟩	🟥		
ii. Analyse and categorise requriements		7 hours each		🟩	🟥		
iii. Prioritise requirements		4 hours each			🟥		
iv. Design and mockups of:		5 hours each			🟥		
a. System architecture		3 hours each			🟥		
b. Data Modeling	Kai Wen	2 hours			🟥		
c. Database schema design	Candice	4 hours			🟥		
d. User interface Design	All members	7 hours each			🟥		
v. Preparation of documentation for submission		10 hours each		🟩	🟩	🟩	🟥

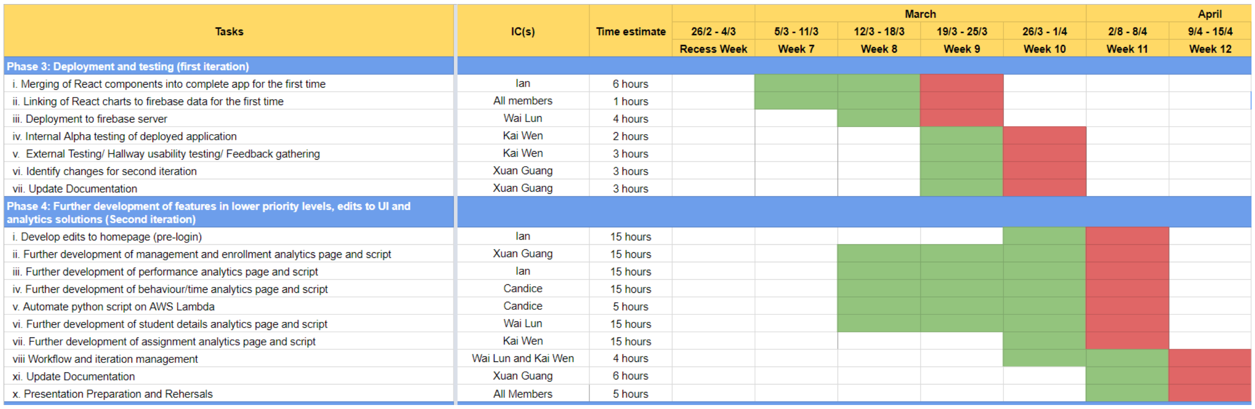
Phase 2: Preliminary UI and analytics development(first iteration, develop about ¾ of priority 1 features.



Tasks	IC(s)	Time estimate	February			March		
			12/2 - 18/2	19/2 - 25/2	26/2 - 4/3	5/3 - 11/3	12/3 - 18/3	19/3 - 25/3
			Week 5	Week 6	Recess Week	Week 7	Week 8	Week 9
Phase 2: Preliminary UI and analytics development (First iteration)								
i. Development of homepage (pre-login) and login page	Ian	20 hours			🟩	🟥		
ii. Development of management and enrollment analytics page	Xuanguang	20 hours				🟩	🟩	🟥
iii. Development of python analytics script for management and enrollment analytics						🟩	🟩	🟥
iv. Development of performance analytics page	Ian	20 hours				🟩	🟩	🟥
v. Development of python analytics script for Performance analytics						🟩	🟩	🟥
vi. Development of behaviour/time analytics page	Candice	20 hours				🟩	🟩	🟥
vii. Development of python analytics script for Behaviour/time analytics						🟩	🟩	🟥
viii. Development of student details page	Wai Lun	20 hours				🟩	🟩	🟥
ix. Development of python analytics script for student details page						🟩	🟩	🟥
x. Development of assignment details page	Kai Wen	20 hours				🟩	🟩	🟥
xi. Development of python analytics script for assignment details page						🟩	🟩	🟥
xii Workflow and iteration management	Wai Lun and Kai Wen	4 hours				🟩	🟩	🟥

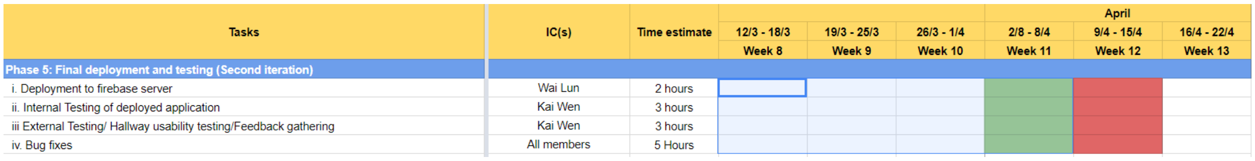
Phase 3 Deployment and testing of first iteration

And Phase 4 Further development of remaining features kept (Second iteration)



Tasks	IC(s)	Time estimate		March				April	
			26/2 - 4/3	5/3 - 11/3	12/3 - 18/3	19/3 - 25/3	26/3 - 1/4	2/8 - 8/4	9/4 - 15/4
			Recess Week	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Phase 3: Deployment and testing (first iteration)									
i. Merging of React components into complete app for the first time	Ian	6 hours		🟩	🟩	🟥			
ii. Linking of React charts to firebase data for the first time	All members	1 hours		🟩	🟩	🟥			
iii. Deployment to firebase server	Wai Lun	4 hours			🟩	🟥			
iv. Internal Alpha testing of deployed application	Kai Wen	2 hours				🟩	🟥		
v. External Testing/ Hallway usability testing/ Feedback gathering	Kai Wen	3 hours				🟩	🟥		
vi. Identify changes for second iteration	Xuan Guang	3 hours				🟩	🟥		
vii. Update Documentation	Xuan Guang	3 hours				🟩	🟥		
Phase 4: Further development of features in lower priority levels, edits to UI and analytics solutions (Second iteration)									
i. Develop edits to homepage (pre-login)	Ian	15 hours					🟩	🟥	
ii. Further development of management and enrollment analytics page and script	Xuan Guang	15 hours			🟩	🟩	🟩	🟥	
iii. Further development of performance analytics page and script	Ian	15 hours			🟩	🟩	🟩	🟥	
iv. Further development of behaviour/time analytics page and script	Candice	15 hours			🟩	🟩	🟩	🟥	
v. Automate python script on AWS Lambda	Candice	5 hours			🟩	🟩	🟩	🟥	
vi. Further development of student details analytics page and script	Wai Lun	15 hours			🟩	🟩	🟩	🟥	
vii. Further development of assignment analytics page and script	Kai Wen	15 hours					🟩	🟥	
viii Workflow and iteration management	Wai Lun and Kai Wen	4 hours					🟩	🟩	🟥
xi. Update Documentation	Xuan Guang	6 hours						🟩	🟥
x. Presentation Preparation and Rehersals	All Members	5 hours						🟩	🟥

Phase 5 Final deployment and testing of second iteration



Tasks	IC(s)	Time estimate				April		
			12/3 - 18/3	19/3 - 25/3	26/3 - 1/4	2/8 - 8/4	9/4 - 15/4	16/4 - 22/4
			Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
Phase 5: Final deployment and testing (Second iteration)								
i. Deployment to firebase server	Wai Lun	2 hours				🟩	🟥	
ii. Internal Testing of deployed application	Kai Wen	3 hours				🟩	🟥	
iii External Testing/ Hallway usability testing/Feedback gathering	Kai Wen	3 hours				🟩	🟥	
iv. Bug fixes	All members	5 Hours				🟩	🟥	

Aside from the above tasks we have also taken on the following responsibilities for the duration of the project:

Role	Team member
Product Management (strategy, gather feedback and testing, features and release management)	Kai Wen
Consolidating React components	Ian
Project Management (Timeline, scope management, resource management)	Wai Lun
UI Management	Wai Lun & Ian
Formatting and updating documentation, Change log.	Xuan Guang
Consolidate Python scripts and maintaining the database	Candice

04.08 UI Design (Old)

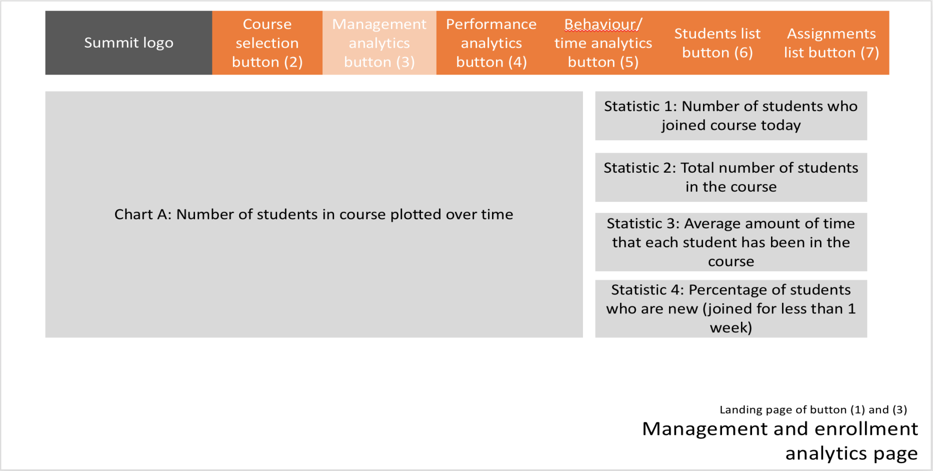
04.08.01 High Fidelity Mockups

We designed a series of high-fidelity mockups that illustrate our application interface and the key features that will be in the application for iteration 1. We have centred our application on providing analytics surrounding a particular course that an instructor might be interested in. As such, all the mockups below illustrate features that an instructor might need in the management of a course.

Mockup 1: Course management and enrollment analytics page

Corresponds to functional requirement 1(i)(a),1(ii), and workplan phase 2 ii) and iii)

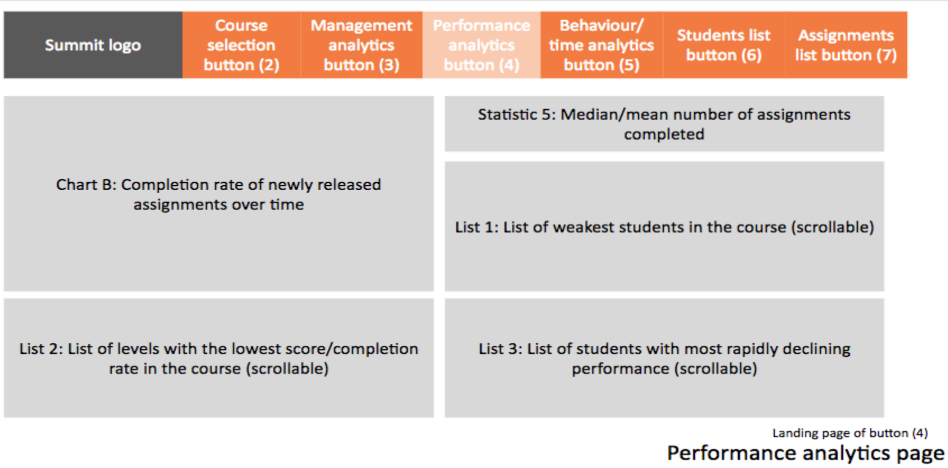
See Prioritisation (04.07.01) and Project Work Plan (04.07.02) sections for more information


Summit logo
Course selection button (2)
Management analytics button (3)
Performance analytics button (4)
Behaviour/ time analytics button (5)
Students list button (6)
Assignments list button (7)
Chart A: Number of students in course plotted over time
Statistic 1: Number of students who joined course today
Statistic 2: Total number of students in the course
Statistic 3: Average amount of time that each student has been in the course
Statistic 4: Percentage of students who are new (joined for less than 1 week)
Landing page of button (1) and (3)
Management and enrollment analytics page

Mockup 2: Performance analytics page

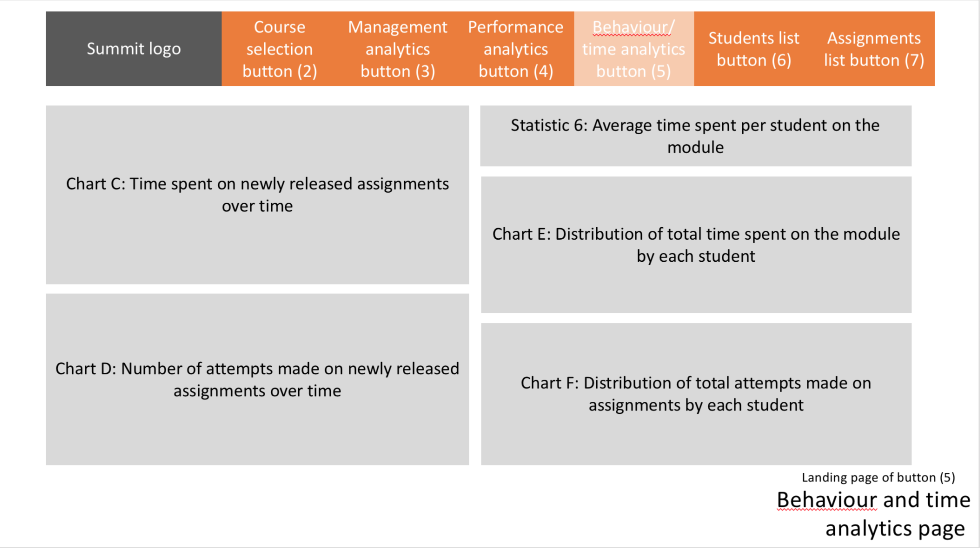
Corresponds to functional requirement 2(iii)(a), and workplan phase 2 iv) and v)

See Prioritisation (04.07.01) and Project Work Plan (04.07.02) sections for more information


Summit logo
Course selection button (2)
Management analytics button (3)
Performance analytics button (4)
Behaviour/ time analytics button (5)
Students list button (6)
Assignments list button (7)
Chart B: Completion rate of newly released assignments over time
Statistic 5: Median/mean number of assignments completed
List 1: List of weakest students in the course (scrollable)
List 2: List of levels with the lowest score/completion rate in the course (scrollable)
List 3: List of students with most rapidly declining performance (scrollable)
Landing page of button (4)
Performance analytics page

Mockup 3: Behaviour and time analytics page

Corresponds to functional requirement 3(iii), and workplan phase 2 vi) and vii)
See Prioritisation (04.07.01) and Project Work Plan (04.07.02) sections for more information



Summit logo

Course selection button (2)

Management analytics button (3)

Performance analytics button (4)

Behaviour/ time analytics button (5)

Students list button (6)

Assignments list button (7)

Chart C: Time spent on newly released assignments over time

Statistic 6: Average time spent per student on the module

Chart E: Distribution of total time spent on the module by each student

Chart D: Number of attempts made on newly released assignments over time

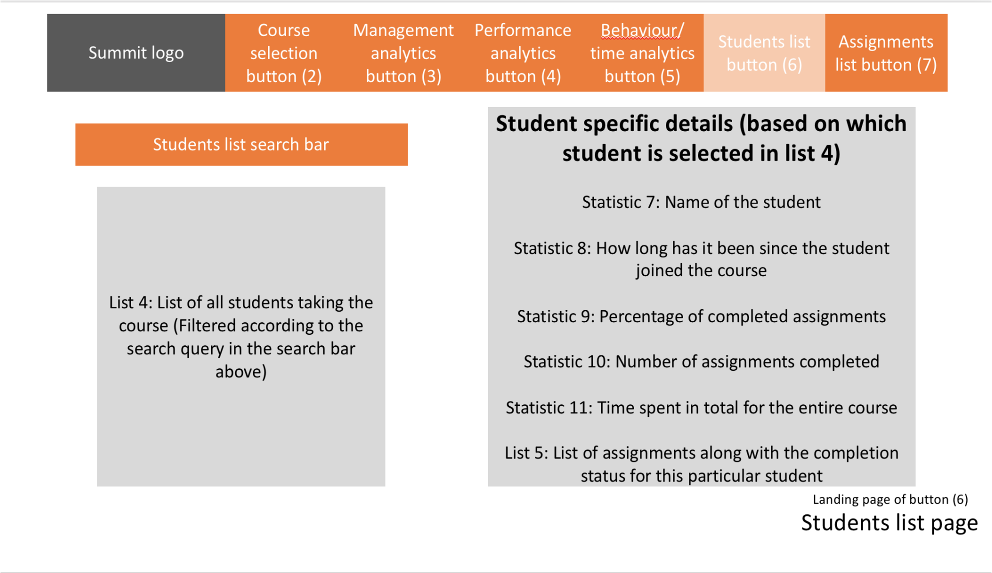
Chart F: Distribution of total attempts made on assignments by each student

Landing page of button (5)

Behaviour and time analytics page

Mockup 4: Students list page

Corresponds to functional requirement 2(i)(a), 3(ii), and workplan phase 2 viii) and ix)
See Prioritisation (04.07.01) and Project Work Plan (04.07.02) sections for more information



Summit logo

Course selection button (2)

Management analytics button (3)

Performance analytics button (4)

Behaviour/ time analytics button (5)

Students list button (6)

Assignments list button (7)

Students list search bar

List 4: List of all students taking the course (Filtered according to the search query in the search bar above)

Student specific details (based on which student is selected in list 4)

Statistic 7: Name of the student

Statistic 8: How long has it been since the student joined the course

Statistic 9: Percentage of completed assignments

Statistic 10: Number of assignments completed

Statistic 11: Time spent in total for the entire course

List 5: List of assignments along with the completion status for this particular student

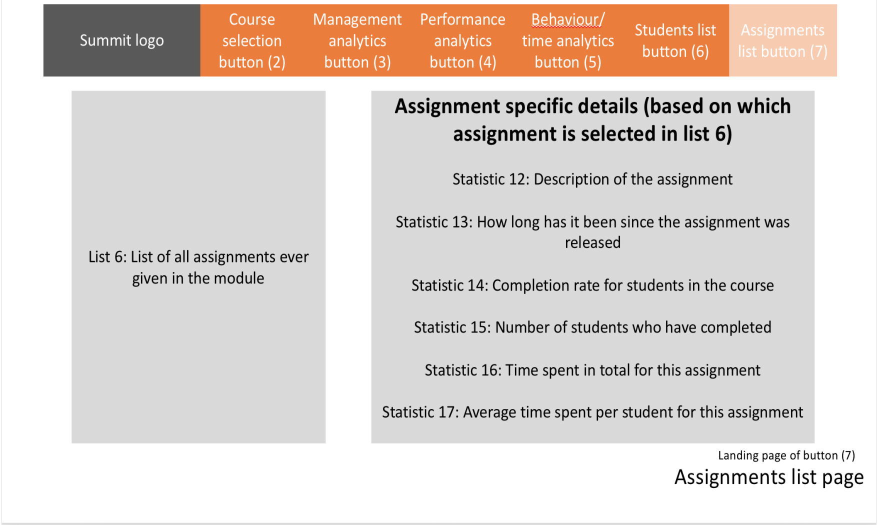
Landing page of button (6)

Students list page

Mockup 5: Assignments list page

Corresponds to functional requirement 2(ii)(a), 3(i), and workplan phase 2 x) and xi)

See Prioritisation (04.07.01) and Project Work Plan (04.07.02) sections for more information


Summit logo
Course selection button (2)
Management analytics button (3)
Performance analytics button (4)
Behaviour/ time analytics button (5)
Students list button (6)
Assignments list button (7)

List 6: List of all assignments ever given in the module

Assignment specific details (based on which assignment is selected in list 6)

Statistic 12: Description of the assignment

Statistic 13: How long has it been since the assignment was released

Statistic 14: Completion rate for students in the course

Statistic 15: Number of students who have completed

Statistic 16: Time spent in total for this assignment

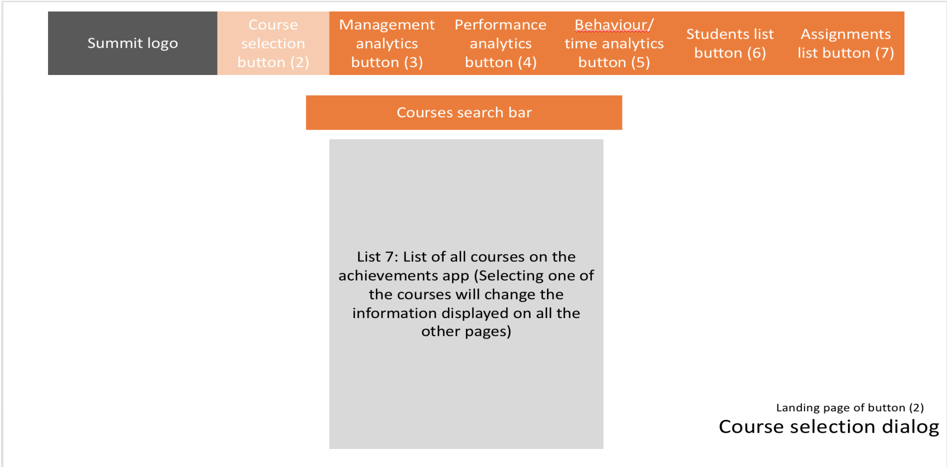
Statistic 17: Average time spent per student for this assignment

Landing page of button (7)

Assignments list page

Mockup 6: Course selection dialog

This dialog enables instructors to select the course that he/she wants to understand more about


Summit logo
Course selection button (2)
Management analytics button (3)
Performance analytics button (4)
Behaviour/ time analytics button (5)
Students list button (6)
Assignments list button (7)
Courses search bar

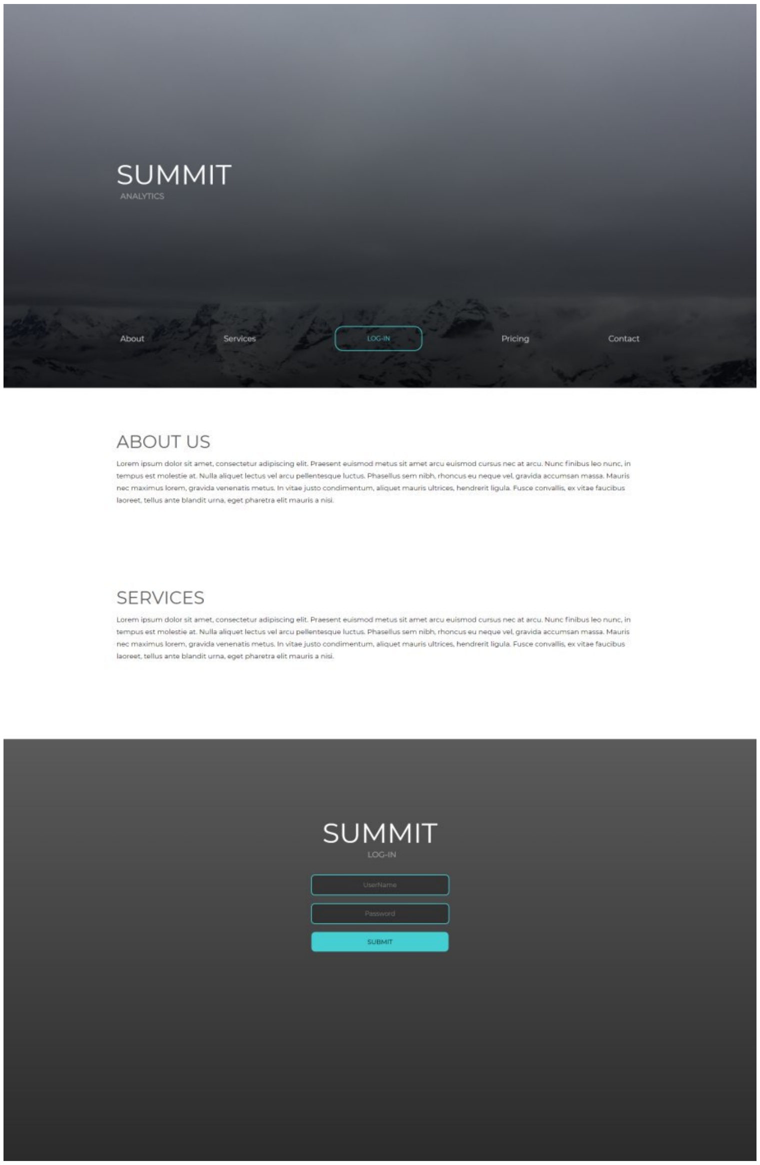
List 7: List of all courses on the achievements app (Selecting one of the courses will change the information displayed on all the other pages)

Landing page of button (2)

Course selection dialog

04.08.02 UI Design Schema

To give ourselves a clearer picture of how the app would look like, and also to provide a framework for the team to begin building the relevant React components, we created the following UI mockups. We have functioning buttons on the mockup and have also managed to import a number of rechart components to ensure that it is working properly. The github repository for our preliminary app can be found at the following link: https://github.com/TheIanSim/summit-on-react/tree/master/src

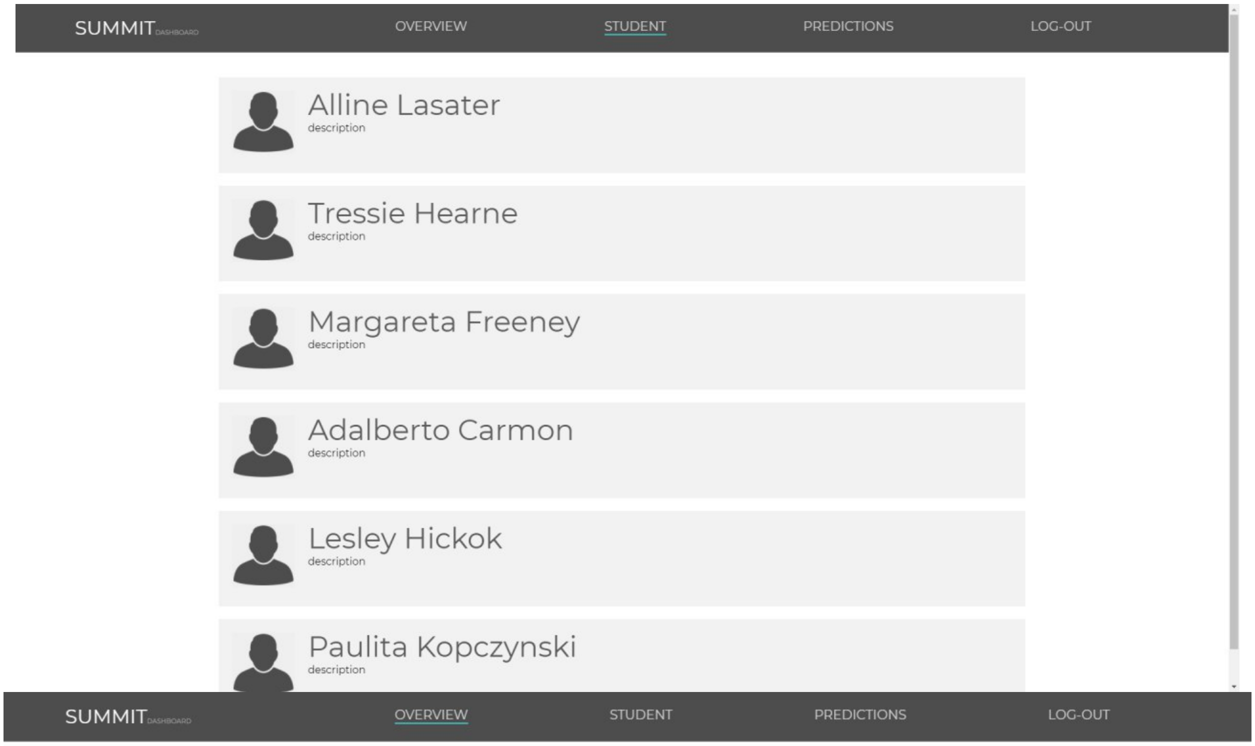


ABOUT US

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod metus sit amet arcu euismod cursus nec at arcu. Nunc finibus leo nunc, in tempus est molestie at. Nulla aliquet lectus vel arcu pellentesque luctus. Phasellus sem nibh, rhoncus eu neque vel, gravida accumsan massa. Mauris nec maximus lorem, gravida venenatis metus. In vitae justo condimentum, aliquet mauris ultrices, hendrerit ligula. Fusce convallis, ex vitae faucibus laoreet, tellus ante blandit urna, eget pharetra elit mauris a nisi.

SERVICES

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod metus sit amet arcu euismod cursus nec at arcu. Nunc finibus leo nunc, in tempus est molestie at. Nulla aliquet lectus vel arcu pellentesque luctus. Phasellus sem nibh, rhoncus eu neque vel, gravida accumsan massa. Mauris nec maximus lorem, gravida venenatis metus. In vitae justo condimentum, aliquet mauris ultrices, hendrerit ligula. Fusce convallis, ex vitae faucibus laoreet, tellus ante blandit urna, eget pharetra elit mauris a nisi.



SUMMIT DASHBOARD

OVERVIEW STUDENT PREDICTIONS LOG-OUT

Alline Lasater
description

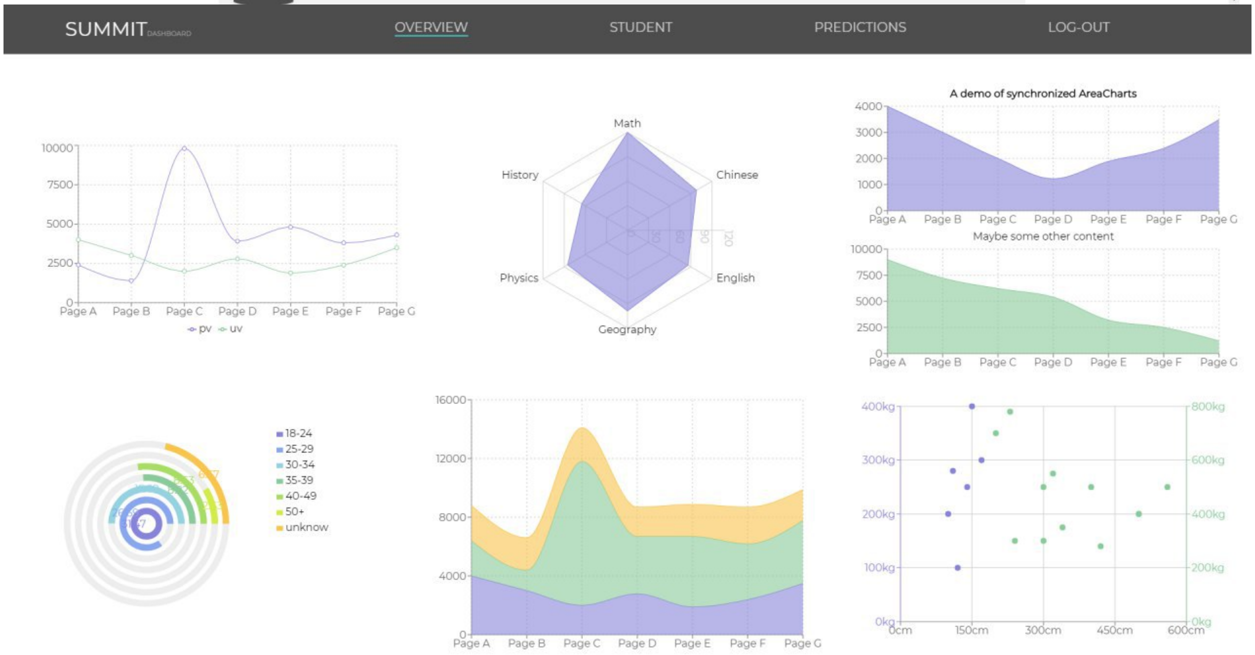
Tressie Hearne
description

Margareta Freeney
description

Adalberto Carmon
description

Lesley Hickok
description

Paulita Kopczynski
description



SUMMIT DASHBOARD

OVERVIEW STUDENT PREDICTIONS LOG-OUT