



Lerni & Dividi

SRS Project

Centennial Comp 225 - 007

Summer Project

Marianne Palmer, Shota Ito,

Haris Siddiqi, Thiago Batista

Section 1. Introduction	2
1.1 Purpose	2
1.2 Document Conventions	2
1.3 Intended Audience and Reading Suggestions	2
1.4 Project Scope:	2
1.5 References	3
Section 2. General System Description	4
2.1 Product Perspective	4
2.2 Product Features	4
2.3 User Classes and Characteristics	4
2.4 Operating Environment:	5
Section 3. System Capabilities, Conditions, and Constraints	5
3.2 Hardware interfaces	5
3.3 Software Interfaces	6
Appendix C	6
Appendix D	7
Functional Requirements List	10
Nonfunctional Requirements List	11
Use Case Table	13
Use Case for Template for Messenger	15
Visio use case for public chat diagram	18
Swim Lane Diagram	19
Appendix E	20
Class Diagram	20
State Diagrams	25

Section 1. Introduction

1.1 Purpose

The new software will address the problem of student networking and interaction during online classes. It will provide a forum for them to interact with each other so they can spread the knowledge in a friendly stress-free environment.

1.2 Document Conventions

There are a few conventions used in this document. Please see the table below for reference.

Acronyms	Description
SQL	Structured query language
Project	Lerni and Dividi App
API	Application programming interface
DBMS	Database Management System
UI	User Interface
UX	User Experience

1.3 Intended Audience and Reading Suggestions

This document is intended for staff including developers, investors, marketing officers, and managers to help them better understand the product.

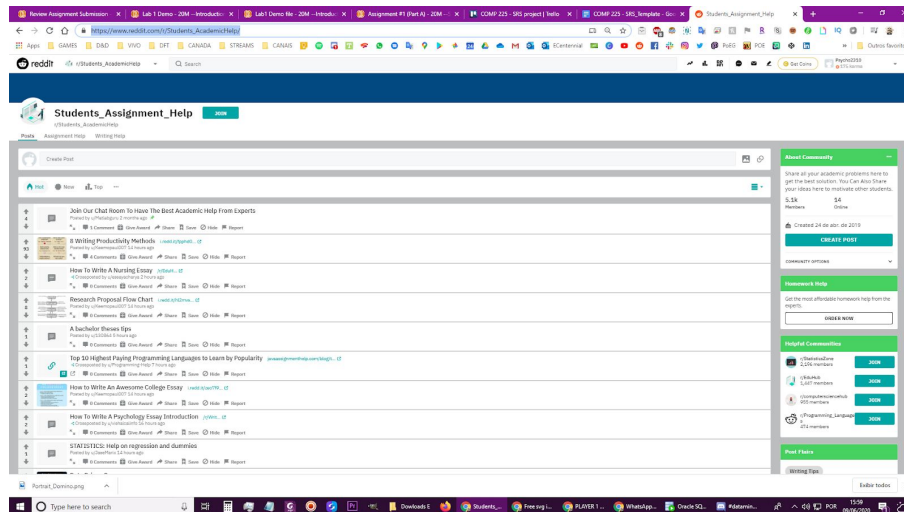
1.4 Project Scope:

Online learning has proved that there is an increased need for a means of interaction for students. While some colleges and universities have their own learning platforms, such as D2L, these are overseen by faculty members and can discourage casual and open communication between students. Our application is a student-to-student platform which will create a non-restrictive environment for students; therefore, it will facilitate improved cooperation, collaboration and peer mentorship. Through this app, students will have the ability to add or join a class (e.g. COMP225 Spring), take part in discussions and update the class calendar with due dates. Our app will be completely moderated by the students. It will also have a student

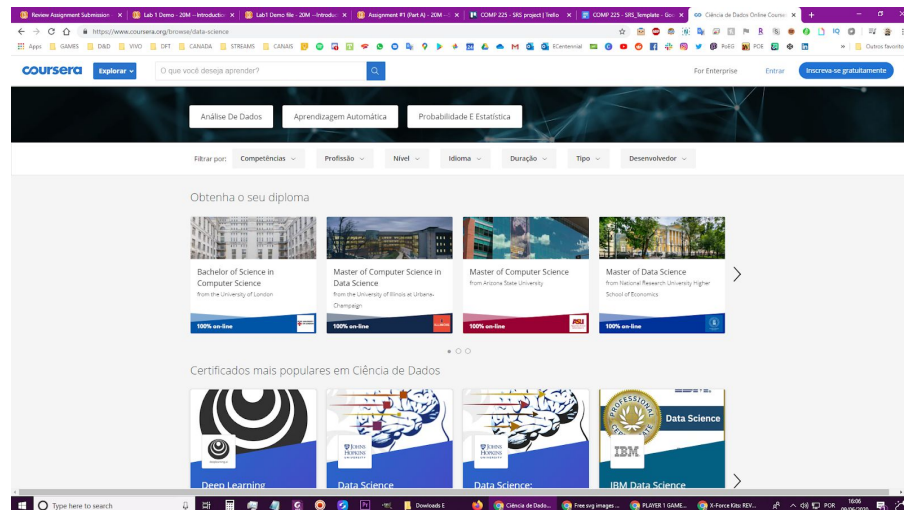
verification process in the form of security questions (submitted by the moderator) which will ensure that all class participants are authentic.

1.5 References

Reddit Students Academic Help Thread - https://www.reddit.com/r/Students_AcademicHelp/



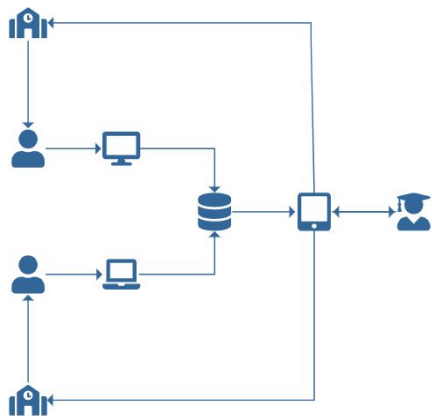
Coursera Data Science Online Courses - <https://www.coursera.org/browse/data-science>



Section 2. General System Description

2.1 Product Perspective

This is a new open-source product, under the General Public License. It is a Web-Based System implemented on a client-server model that supports Cross Platforming with mobile devices and hence will require for both Internet connection to fetch and display results. The Lerni Dividi System provides a simple mechanism for users to share and acquire knowledge between them; to do this, the application will need to connect to remote servers that will communicate with an oracle database server.



After their classes students will enter the application's platform, accessing the database chats related to class forums. A student enrolled as the forum moderator will have access to the class' list, so he can ensure that only students are logging in the public forums.

2.2 Product Features

The app will perform the following actions:

1. Allow students to talk to each other in group chats
2. Ask questions on the class forum
3. Assign moderators to each class to verify participants and moderate messages
4. Allow students to coordinate class schedules for assignments and tests.

2.3 User Classes and Characteristics

Students: will use the software to connect to their peers in their online classes.

Moderators: Will use the software to check the users are students enrolled with the colleges, also participate in their online classes, and upload support materials

2.4 Operating Environment:

1. **Database:** MongoDB
2. **Browser:** Chrome, Firefox, and Safari (most students are using iPhone. So the software has to be compatible with Safari)
3. **Developing language:** JavaScript

2.7 Assumptions and Dependencies:

We assume the following:

1. Revenue will come from ads and/or donations, or premium upgrades.
2. All open-source technologies that will be used in the development will stay open source.
3. Users will have a good internet connection.
4. Users are familiar with using internet browsers and are adept at handling the keyboard and mouse.

Section 3. System Capabilities, Conditions, and Constraints

3.1 User Interfaces:

Users will access the system through their web browser.

The frameworks that will be used are:

- **Front-end:** React
- **Back-end:** Node.js

3.2 Hardware interfaces

The app will connect to the following hardware devices:

- Camera
- Mobile phone
- Microphone

3.3 Software Interfaces

1. SSO (Single Sign-On) - Used for authenticating users
2. Calendar (for sharing class schedule)
3. Phone notifications
4. Ads or Donation (source of funding): Youtube ads and Google AdSense
5. Facebook API
6. MailChimp Transactional Email API
7. Paypal API

Appendix C

Stakeholder Register					
Stakeholder Name	Stakeholder Position	External /Internal	Stakeholder contact details	Operational / Executive	Interest (High, Medium, Low)
Andy Weir	Customer	External	andy.weir@gmail.com	Operational	High
Ben Wyatt	Customer - Moderator	External	b.wyatt@gmail.com	Operational	High
John Scalzi	Software Developer	Internal	j.scalzi@lernidividi.com	Executive	Medium
Neil Gaiman	Investor	External	neil.gaiman@gmail.com	Executive	High
Andre Brown	Marketing Officer	Internal	a.brown@lernidividi.com	Operational	Medium
Ray Arnold	Chief Technology Officer (CTO)	Internal	r.arnold@lernidividi.com	Executive	High
Bobby Axelrod	Chief Financial Officer (CFO)	Internal	b.axelrod@lernidividi.com	Executive	High

Appendix D

Interview Questions		
Question	Stakeholder position	Answer
When would you like to use this App?	Customer	<i>"When studying, especially before an exam or an assignment."</i>
How would you like to install this app?	Customer	<i>"I would like to install the App directly from my iPhone or Android store."</i>
Where would you like to find this app?	Customer	<i>"It would be nice to have it both on the App store and College website."</i>
How would you like to login?	Customer	<i>"I found that it is easier when I login using my Facebook account, so that would be a great feature."</i>
How often would you use this app?	Customer	<i>"I think around 3-5 Days a week, for no more than one hour each day."</i>
How will you make the UI and UX appealing to our target audience?	Software Developer	<i>"Since the target audience is college/university students I will use eye-catching colors to grab attention. For the design, I will choose a minimal and functional interface to make it easy to use."</i>
How would you implement a menu bar and a login page?	Software Developer	<i>"I would put the company logo on the main page. Assuming customers use this app on smartphones, many people are right-handed, so the menu bar will be top right."</i>
What technologies would you suggest to use for managing the database?	Software Developer	<i>"This is a large application, so it will have a lot of data to manage. I suggest we use MongoDB as the database management tool. Also, we should use AWS for the servers."</i>

How would you work with other members of our team?	Software Developer	<p><i>"For the technical team, I will use different tools such as scrum meetings, slack, and Trello."</i></p> <p><i>For non-technical people, I always try to explain the technical aspects of the project in a way that they can understand. I also value their input and feedback and make sure to relay that back to the technical team."</i></p>
What version control software would you use for this project?	Software Developer	<i>"I am most comfortable with Git. It is an amazing tool and I believe it has everything we are going to need for this project."</i>
How much are you willing to invest?	Investor	<i>"I'm willing to invest \$100,000 in this first phase and then another \$200,000 once we are consolidated in the market."</i>
Will you allocate the full amount right from the start?	Investor	<i>"Yes, I will. I already got the money allocated in a separate bank account."</i>
Is there anything that is mandatory for you in this project?	Investor	<i>"Since this is going to be a free app, I would like some kind of 'premium' feature so I can get some money back."</i>
Is it possible to invest again if we got success?	Investor	<i>"Yes, I will, but it depends on the return rate of this first phase."</i>
What return on this investment are you expecting?	Investor	<i>"Like with many start-ups, I don't expect to see much of a return in the first year. However, after this period, I strongly feel that this app will be profitable."</i>
What is going to be your marketing strategy plan for this app?	Marketing Officer	<i>"Through an internet marketing campaign specifically targeted for students. We will use a combination of Youtube ads and Google AdSense."</i>

How will you specifically target our audience of students?	Marketing Officer	<i>"I will tailor the ads to the platforms and websites that students use."</i>
Do you think it marketing is going to be important for this project?	Marketing Officer	<i>"Definitely. While word of mouth will help the app grow, that growth will be slow. Through properly marketing our app, we can quickly reach out to a large audience."</i>
Do you think marketing this app will be difficult?	Marketing Officer	<i>"No, I don't. We will use the best strategies available to us of course. But I feel that there is a high demand for a platform like this and students will naturally gravitate to it."</i>
Is there anything that we should implement to make the marketing easier?	Marketing Officer	<i>"Please try to make it as user friendly as possible, as few options as possible to get where they want to."</i>
What additional features would you like in the next release?	Chief Technology Officer	<i>"I would love to see a chat feature implemented. It is going to be crucial to moving our project closer to its final version."</i>
From your research, are there any new technologies that can be used in this app?	Chief Technology Officer	<i>"Currently I believe it best if we use stable technologies such as node.js and MongoDB as they are secure and reliable. However, I keep myself up to date with the market and if there are any new technologies that would be a great fit for our app I will definitely consider them."</i>
What is the timeline for the app's development?	Chief Technology Officer	<i>"Approximately 5.5 months for the complete app. 2 weeks for planning, 2 weeks for design, 4 months for frontend and backend development and 2 weeks for testing. 1 week for finalization and launch."</i>
Are there any development frameworks that you would like to use?	Chief Technology Officer	<i>"I expect to use the Scrum framework."</i>
How will your previous	Chief Financial Officer	<i>"My previous experiences working</i>

experience help you in our company?		<i>for larger organizations will help me take this startup to the next level and guide it in the right direction."</i>
What is your long term vision for this organization?	Chief Financial Officer	<i>"I see this company being a leader in the niche market of social networks in the education sector."</i>
What is your short term vision for the organization?	Chief Financial Officer	<i>"Right now the priority is to get a beta version of the app out there so we can get feedback and move closer to our goal."</i>
How will you manage the company's budget?	Chief Financial Officer	<i>"I have an excellent accounting team under me that will help me keep the budget on track"</i>
What is the ideal number of employees for an organization like ours?	Chief Financial Officer	<i>"At this stage, we are a small company and need to ensure we are conserving resources and only keeping essential workers on payroll."</i>

Functional Requirements List

Functional Requirements List				
Requirement ID	Requirement Title	Short Description	Priority	Requester
FR01	Dashboard	The app will display all classes that a student is enrolled in along with the navigation.	High	Marketing Officer
FR02	Ads	The app will generate revenue from ads: Youtube ads and Google AdSense.	High	Marketing Officer
FR03	Public chat	The app allows users to publicly communicate with each other.	High	Student
FR04	Private chat	The app allows users to	High	Student

		privately communicate with each other.		
FR05	Premium	The app allows users to upgrade to a premium account.	High	Investor
FR06	Alerts	The app will alert users whenever there is a new message or calendar reminder.	Medium	Student
FR07	@ reply	The app provides priority notification to targeted users.	Medium	Student
FR08	Customize profile	The app allows users to customize their profile.	Medium	Student
FR09	Customize forum design	The app allows users to customize the forum logo, color, etc.	Low	Student
FR10	FAQ	The app provides setup instructions and answers common questions.	Low	Manager
FR11	Calendar	The app will have a calendar to share due dates for assignments and exams.	Low	Student

Nonfunctional Requirements List

Nonfunctional Requirements List				
Requirement ID	Requirement Title	Short Description	Priority	Requester
NFR01	Web browser compatibility	The system should run on all web browsers including Chrome, Firefox, and Safari.	High	Student
NFR02	User Experience	The UI will be clean, simple and intuitive to ensure	High	Marketing Officer

		optimal user experience.		
NFR03	Performance	There should be a less than 2-second delay between the time a message is sent and the time it is received.	High	Student
NFR04	Maintainability	The development team will use GitHub for source control and make the app should be flexible enough to go through changes to the code with ease.	High	Developer
NFR05	Functionality	The app should be practical. There shouldn't be any unnecessary/unrequested features or functions. This will create a streamlined experience for the user.	High	Marketing Manager
NFR06	Operation	The app will be compatible with both high-end and low-end devices. To ensure this, it will be tested on a broad array of different devices.	Medium	Student
NFR07	Scalability	The app will be divided into modules which should handle scalability from small to large class sizes.	Medium	Developer
NFR08	Security	In order to ensure the validity of users (if they are indeed a member of the class), the moderator will verify participants, through a security question process.	Medium	Developer

Use Case Table

Use Cases			
Use Case name	List of related Requirement id	Actor(s)	Brief Description
Send Message	FR03, FR04	Student Notification API Real-time Messaging API	The student will select the Messenger. Using the Real-time Messaging API, the software app will then present the actor with the 'new message' option. The student is presented with the form to compose the message. The student selects the receiver as 'all'. The student will type a message and post it there. The message will get added to the database. The server will relay the message to all students. All students will be able to reply on public chat (the same method) or private chat (described below). When a message is received, the student will get a notification from the notification API on his/her browser.
Send Private Message	FR04	Student Notification API Real-time Messaging API	The student will select the Messenger. Using the Real-time Messaging API, the software app will then present the actor with the 'new message' option. The student is presented with the form to compose the message. The student will select the 'class-list' button, which will trigger the app to return a list of the students enrolled in the class. The student will then select the person they want to contact another student. The student will type his/her message in the text-box. When the message is complete, the student will click on the 'send message' button. The message will get added to the database. The message will then be delivered by the app to the recipient's inbox. When a message is received, the student will get a notification from the notification API on his/her browser.
Create Post in Forum	FR09	Student Moderator	The moderator can start a new discussion thread in the class form. App will show the option of asking a question, creating a poll, or starting a discussion. All students can continue to make

			posts in the thread until the actor closes the thread.
Purchase Ad Space	FR02	Advertiser Recommendation API Payment API	The app will display options to the advertiser to choose their target demographic and the number of ads they want to purchase. The recommendation API allows them to do this. This allows them to easily create targeted ads on LERNI & DIVIDI, in addition to using Google AdSense. The advertiser will be prompted to pay for the ad space and payment will be handled through the payment API, paypal.
Customize Forum	FR08, FR09	Student	While the student is creating the forum, the app will display a set of 4 different fonts, and a pre-set of 3 colours. It will also give the option to use typographical emphasis (Bold, Italic and Underline) in the forum to customize his discussion thread from. The student will select the options he/she wants and the system will update the forum with the customization.
Create Classroom	FR01	Student Moderator Transactional Email API	From the app's dashboard, the student will select the option 'Create New Classroom'. The student will then be prompted to enter the following information: <ul style="list-style-type: none"> - Course name - Institution name - Date started - List of at least two other class members The Transactional Email API system will then send out an invitation to the other class members to join the class. If the other members join the class, the classroom will be created and the student actor who created the class will become the 'class moderator'.
Add Event to Calendar	FR11	Student Calendar API	The calendar feature is implemented within a class forum. The student selects the calendar option, then the date. The app will display to the student the option to 'add an event'. The student can post an assignment due date, and test and quiz date on the calendar for each class. The student will then select the 'add' button, the event will then be added and displayed to all members of the classroom. The calendar will be able to

			sync events to the student's personal calendar by using a calendar API.
Create Account	FR01	N/A	The user (student or moderator) will enter the link to the Lerni & Dividi website. The log-in page will then be displayed to the actor. Below the login section there will be a button for 'create new account'. The user will select the button and will be prompted to enter his/her email address, a username and a password (which must have at least one uppercase character, one lowercase character and one special character). The password must be entered twice. The user will then select the 'sign-up' button, which will check the validity of the email address and password. If the information is valid, the log-in credentials will be added to the user database.
Login to System	FR01	N/A	The user (student or moderator) will enter the link to the Lerni & Dividi website. The log-in page will then be displayed to the user. The user will enter their username/email and password or facebook account. The user will select the 'login' button. The information entered will then be checked for validity in the database. If the log-in credentials are valid, the user will be signed in. If the log-in credentials are invalid, the user will be prompted to re-enter username/email and password again. If the username/email does not exist in the database, the user will be prompted to create an account.

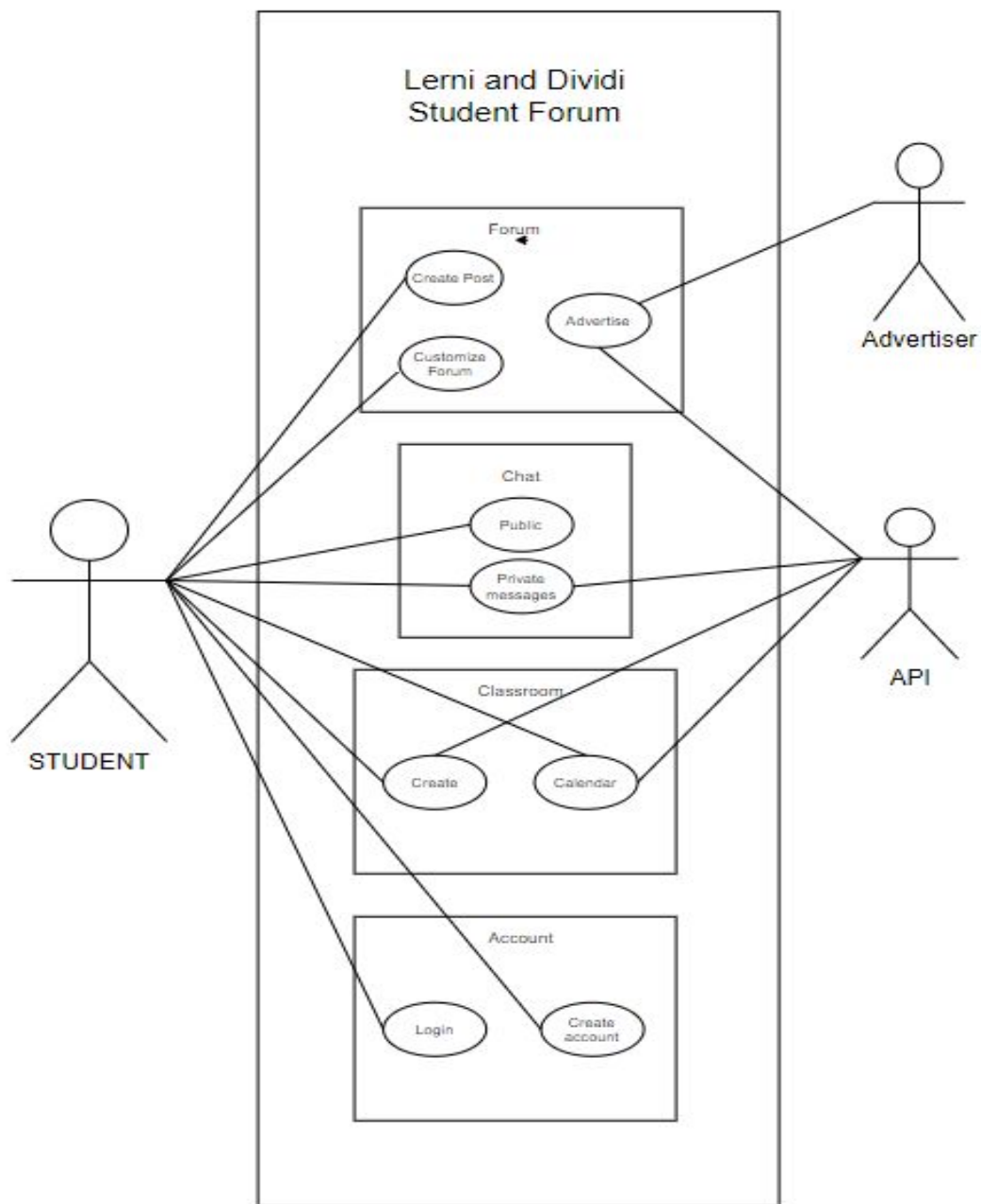
Use Case for Template for Messenger

Use case:	Send Public Message on Messenger
Iterations:	First Iteration: June 29th by R. Arnold.
Primary actor:	Student

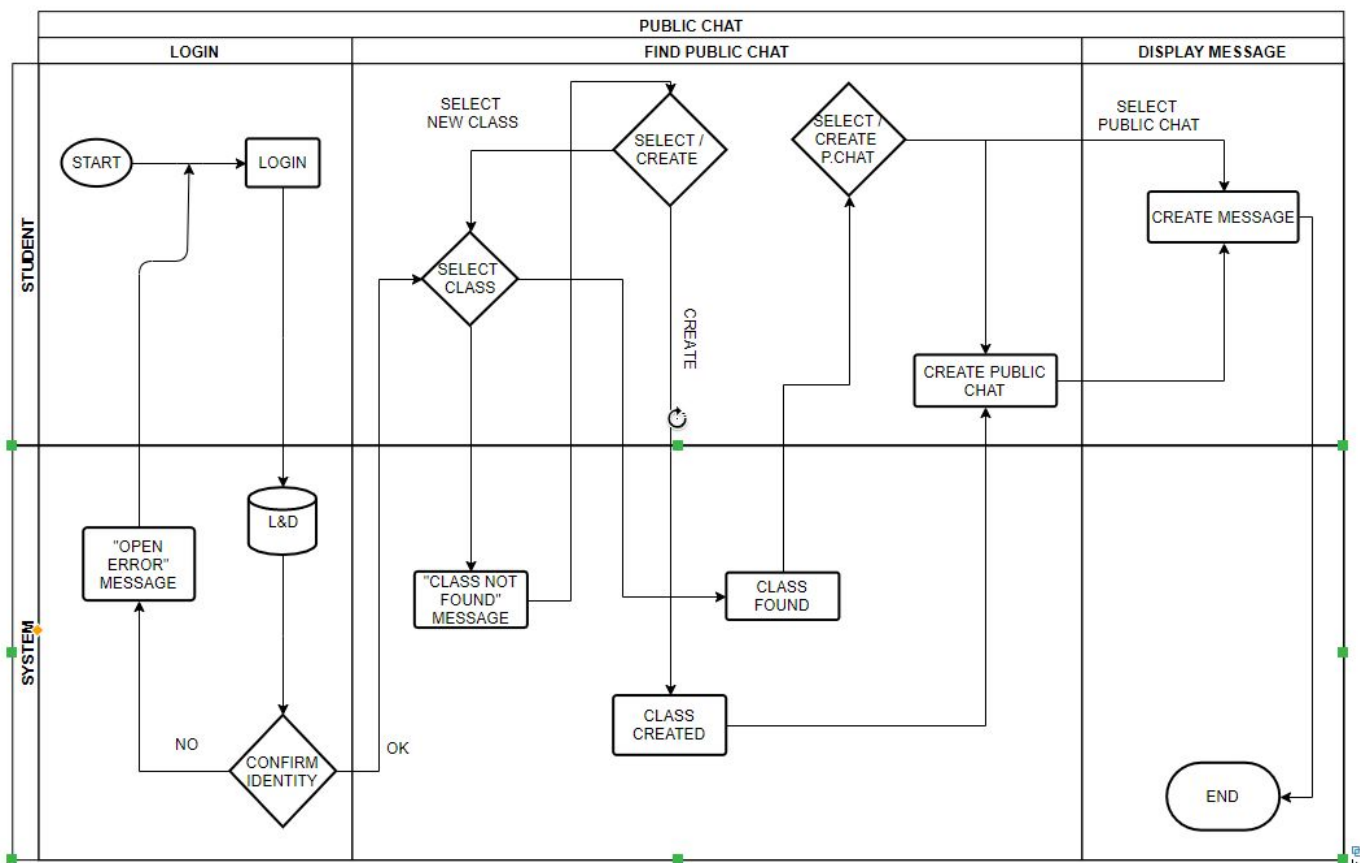
Goal in context:	To send and receive messages publicly on a discussion board page.
Preconditions:	The log-in screen and dashboard must be complete. The student has made an account. A classroom and discussion board has been created.
Trigger:	The student wants to contact all of his/her peers. The student wants to publicly share information.
Scenario:	<ol style="list-style-type: none"> 1. The student logs in to the app 2. The student uses e-mail address and password or facebook account to login. 3. The system displays all frequently used function buttons. 4. The student selects the “Class” button from their function. 5. The student selects the “Messenger” button 6. The student selects target audience as “all” 7. The system enables the student to type any message. 8. After type a message, the student clicks the “Send message” button. 9. The public chat displays the student name, id, date, and message.
Exceptions:	<ol style="list-style-type: none"> 1. Message is too long - characters are beyond the max limit. 2. User is not logged in. 3. Classroom not found. - see use case “Create classroom”. 4. User has been removed from the classroom by the moderator. 5. Client/Server loss of connectivity. 6. Server overloaded - system timed out.
Priority:	High priority, to be implemented after account creation and log-in.
When available:	Before the second increments.
Frequency of use:	Frequent
Channel to actors:	Via all web and mobile browsers.
Secondary actors:	<ol style="list-style-type: none"> 1. Notification API 2. Real Time Messaging API

Channels to secondary actors:	<ol style="list-style-type: none">1. Notification API: Web browser2. Real Time Messaging API: Web browser
Open issues:	<ol style="list-style-type: none">1. Is the validation process sufficient to verify that all students are authentic? I.e. have they been verified as being registered users in their institute.2. Will ad space be too large and reduce the UX?3. Will the delay between messages sent and received be less than 2 seconds?4. Will the system work with older web browsers?

Visio use case for public chat diagram



Swim Lane Diagram



Appendix E

Class Diagram

Class: Dashboard	
Description: Homepage when user logs in	
Responsibility:	Collaborator:
Displays user information	People
Displays list of all classes student is enrolled in	Class
Displays ads	Advertiser
Displays Calendar	Calendar

Class: Account	
Description: Base class for all user accounts	
Responsibility:	Collaborator:
Contains information about the user	AccountHandler

Class: AccountHandler	
Description: Creates account	
Responsibility:	Collaborator:
Collects account info	Account
Create an account	Account
Reads account info	Account

Updates account	Account
Upgrades account to premium	Account
Adds advertiser/student/moderator status to Account	Account, Student, Moderator, Advertiser

Class: Student	
Description: All people, enrolled in a class.	
Responsibility:	Collaborator:
View dashboard	Dashboard
Requests to join a classroom	Moderator
Views class list	ClassRoom
Share calendar with class	ClassRoom, Calendar
Create ClassRoom	ClassRoom
Become Moderator (upon ClassRoom creation)	AccountHandler

Class: Message	
Description: Used to create a message with all message details.	
Responsibility:	Collaborator:
Constructs the message	Messenger

Class: Messenger	
Description: Used to handle messages.	
Responsibility:	Collaborator:
Displays messages	Message

Edits messages	Message
Send public message	Message
Send private message	Message

Class: Moderator	
Description: All students, with moderator status.	
Responsibility:	Collaborator:
Deletes classroom discussion board messages and threads	ClassRoom
Accepts/declines requests to join class	Account, ClassRoom
Invites people to join class	Account, ClassRoom
Transfer moderator status to another student	Student

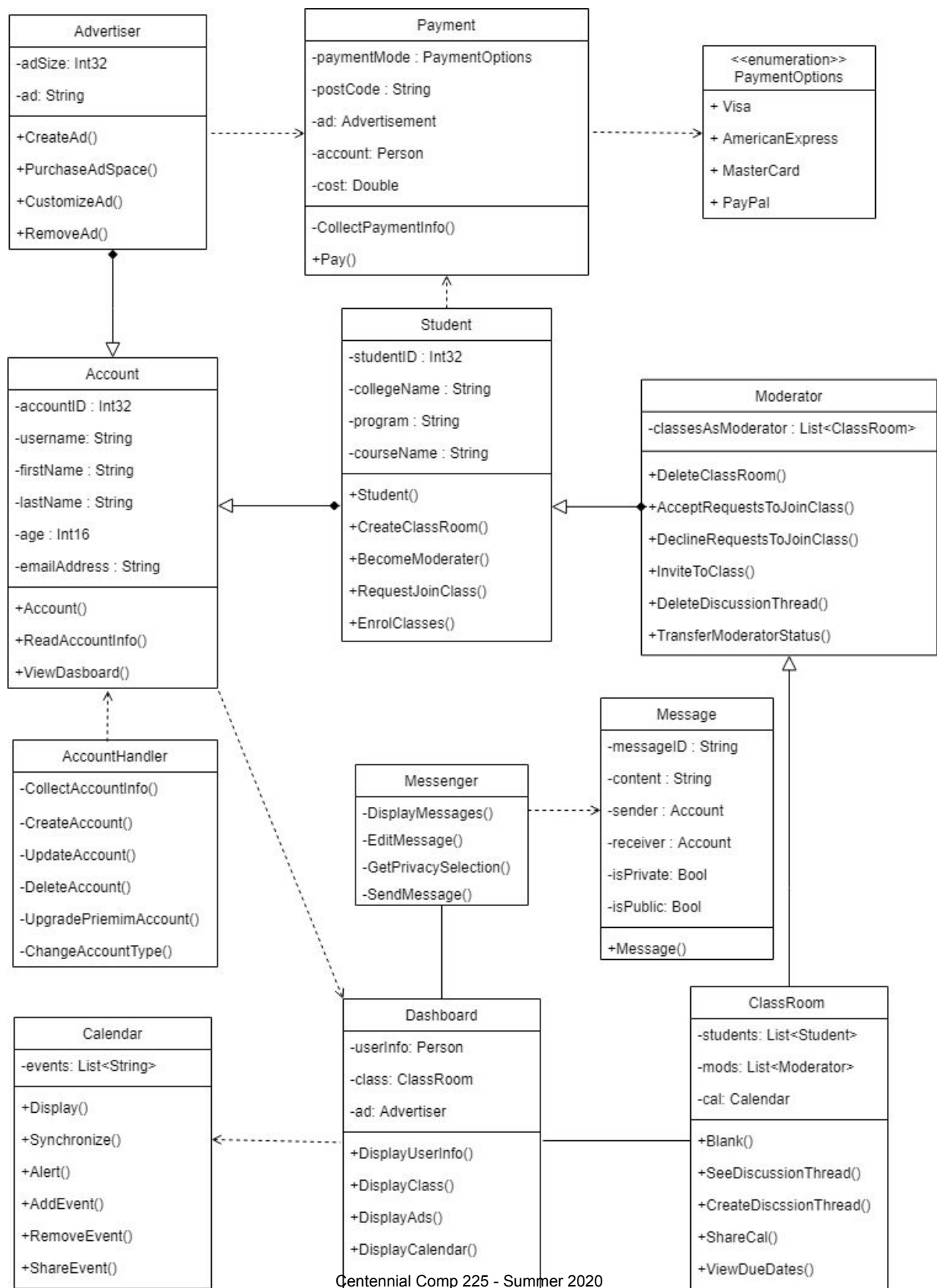
Class: ClassRoom	
Description: Homepage of each class user is in	
Responsibility:	Collaborator:
Share calendar	Calendar
View due dates for class	Calendar
Create discussion thread	
Display thread	

Class: Calendar	
Description: Class calendar accessible from dashboard	
Responsibility:	Collaborator:

Displays selected calendar	
Add events to day	Student
Alerts users of events	People
Synchronizes events with Class	ClassRoom, Student

Class: Advertiser	
Description: Allows user to purchase ads from dashboard	
Responsibility:	Collaborator:
Purchases ad space	Payment
Customize ad	
Creates ad	Dashboard
Remove ad	Dashboard

Class: Payment	
Description: Allows users to make payments	
Responsibility:	Collaborator:
Collect payment information	
Accepts payment for ad space and account upgrade	Advertiser, Account
Accepts some payment options (Visa, Master, paypal)	



State Diagrams

