Design Document

Team 10: Essay Grading System

Atul Aneja, Jalaleldeen Aref, Shivank Tibrewal, Sidhant Chitkara, Tarang Khanna

Table of Contents

Purpose

Today, state departments of education are evolving into new forms of grading and evaluating methods to achieve a more standardized solution for grading. A need for a sophisticated and affordable solution is critical. Since essays are a crucial expression of academic achievement, we are designing an Automated Essay Grading web application. Automating the process of grading essays is extremely useful for teachers, as it is a time consuming and tedious process, with potential for human errors. This tool will also useful for students that would like to grade themselves.

EdX, an online education platform is the closest existing solution. EdX has a system that asks a human grader to evaluate 100 essays, after which it trains itself to grade the same. So the currently available solutions can reduce the long run time constraints, but still require manual grading by each new user to teach the software. Furthermore, existing solutions do not verify the actual data quoted in the essays and seem more one dimensional for grammar and language rather than a complete grader. Existing solutions also use machine learning with human hand labeled features. Our solution will use deep learning and machine learning to find features it deems best. Our solution is to predefine the essay standards and remove individual data entry by users. The final product should provide a grade, alongside a confidence score for that grade.

Design Outline

Overview and Components

The system will implement a client-server architecture with the following components

- Django Web Client
- Django Application Server
- Backend API (containing ML, DL components)
- Firebase DB

Architecture

Our application will use the simple client-server design to provide easy partitions between the components. This way we can easily facilitate the division of development tasks between different team members and allow the the frontend and backend components to evolve in a flexible manner.

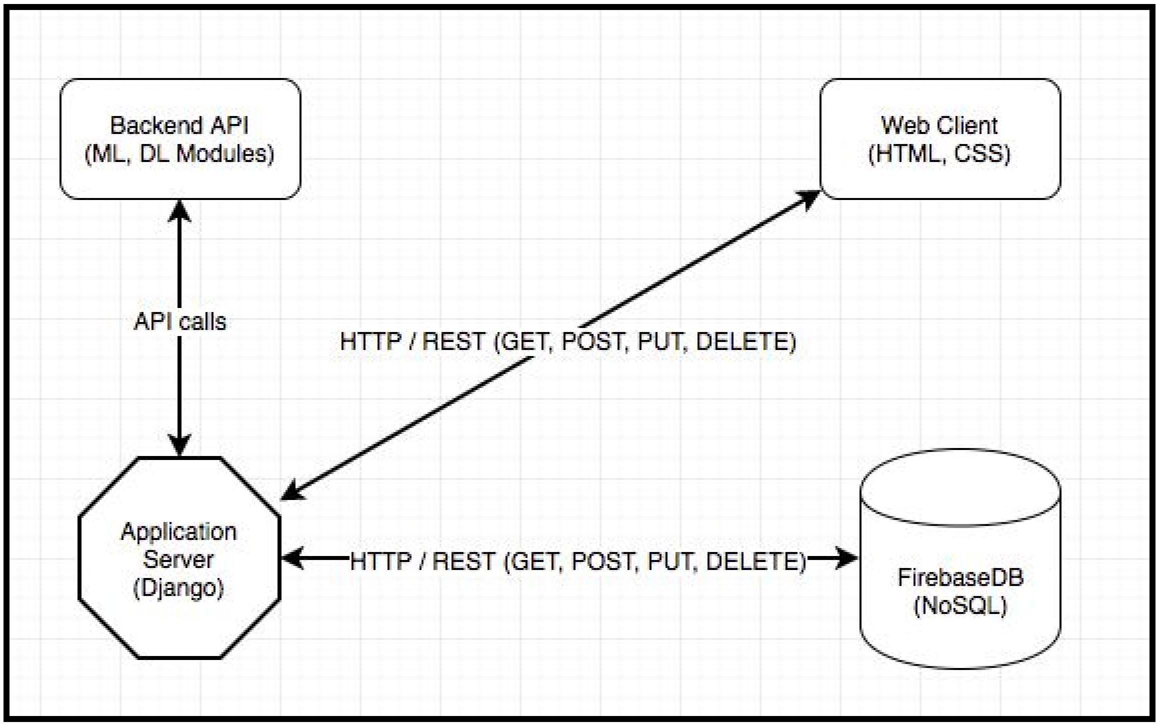


Figure 1: Platform Architecture Diagram

Django Web Client

The django web application will use languages such as HTML, Javascript, CSS and Django template language to construct dynamic web pages that are usable and responsive. This module will be the primary interface between the user and the backend programs. It will send and receive data by interacting with the Django application server.

Django Application Server

The django application server will be responsible for facilitating transfer of data between all the components in our project. It will handle requests by the web client and authenticate users using the DB. This module will be implemented in python.

Backend API

All grading functionality will be implemented in this module and it will be responsible for evaluating essays and then providing a grade to the application server module. This module will be implemented in python and will use frameworks and libraries such as Natural Language Toolkit (NLTK), Scikit Learn, Keras and TensorFlow.

Firebase DB

The database will contain user information and past essay data. The application server module will make calls to this database to authenticate users and retrieve past essay data. To facilitate communication between the two modules we will use a python wrapper called Pyrebase. More information on the wrapper can be found at: https://github.com/thisbejim/Pyrebase

Design Issues

Issue 1

Issue: Which framework to use for the web app:

- Option 1: AngularJS
- Option 2: Flask
- **Option 3: Django**

We decided to use Django for our application server as well as our web client. AngularJS and other similar frameworks were not preferred as cross platform communication with the machine learning backend would be much simpler in Python. Django offers a more conventional structure as opposed to Flask which is suitable as we aim to have a minimalistic front end, with a speedy and intuitive user experience.

Issue 2

Issue: How to take the user's essays

- Option 1: Uploading a PDF
- **Option 2: Uploading a text file**
- **Option 3: Entering text into an inline editor**

We decided to take user input as a text file, any Rich Text Format file or convertible format would suffice. PDF's were the ideal choice because we would not have to worry about different file formats, such as Word documents or Pages files on the mac. PDF's however do not guarantee that input will always be read as input text by the software. Lastly, using an inline editor is easy for the back end but bad for the user as each file would need to be manually opened to grade, so we give this option to users as well.

Issue 3

Issue: How to feed the essay into the classifier

- Option 1: Text
- **Option 2: Word2Vec**
- Option 3: Bag of words

We cannot feed text directly into a classifier, since classifiers only understand in terms of mathematics. So we need to choose a way to convert the text into numbers. Bag of words converts each word into a unique number and stores the frequency for the word. The problem with Bag of words is, it does not capture the relation between different words. Word2vec, converts words into their vector representation. We choose Word2vec since it groups the vectors of similar words together in the vector space.

Issue 4

Issue: Which database software should we use ?

- Option 1: MySQL
- Option 2: Oracle
- **Option 3: Firebase**

Firebase has the simplest queries and is free unlike Oracle. Most of our team has not used SQL extensively so it is easier to use the Firebase provided API. Firebase also handles storing the data online and we do not have to create a server ourselves.

Issue 5

Issue: Which format to store the essays in?

- Option 1: Vectors
- **Option 2: Text file**

Our classifier converts the essays into vector form for prediction, so vector form of storage would be the most efficient in terms of performance and storage. However, it serves as a setback in terms of scalability as we cannot assure that all future algorithms we could employ would use vectors to read data as well, so in order to make the backend expandable in the future, we must also keep the essays stored as text files. During prediction stage of essays we can just convert it to the expected format for the new prediction algorithm.

Design Details

Backend API

An important part of the backend API is training and testing different classifiers such as neural networks, support vector machines and K nearest neighbors and then having the model ready for general predictions. We can know when a classifier is ready by testing its accuracy against the testing data set. This then leads to identifying if one of these classifiers has the highest accuracy or combining the outputs of multiple classifiers produces a better result (ensemble machine learning and voting algorithms). Frameworks like Keras and Tensorflow will help in applying deep neural networks as classifiers. Scikit learn provides access to other classifiers like logistic regression classifier, naive bayes classifier and support vector classifier. NLTK provides helper functions to do natural language processing, such as functions to remove stop words from a text.

A high level view of a classifier's training and testing phase is shown below (Figure 2):

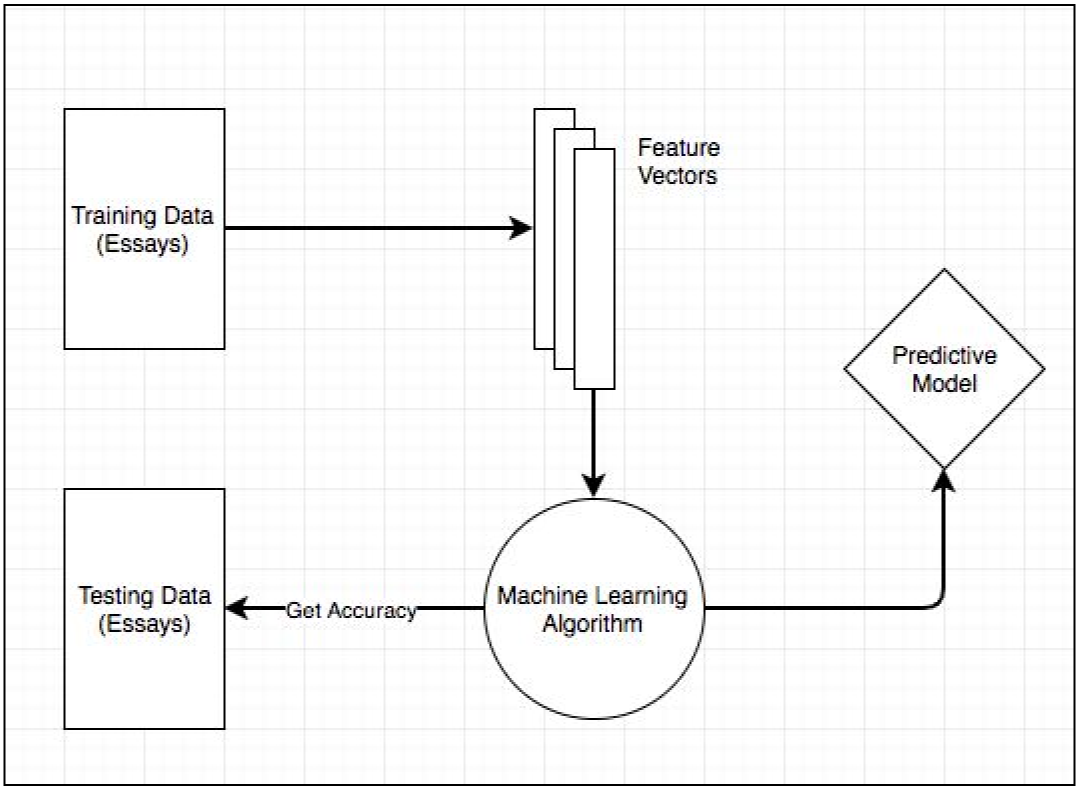


Figure 2: High level view of an algorithm's testing and training phases

Using the above process, we can train multiple classifiers and then filter out those that provide low accuracies. Once we have a set of good (high accuracy) classifiers we can add them to the predictive model. This model can then take an expected grade from each classifier for a given essay. Then we can generate an average grade for all classifiers and send the result back to the user. Based on the differences in grades of each classifier, we can generate a confidence score for the grade as well. This way the user will know how sure the application is of the assigned grade.

Once we sufficiently train all classifiers to predict grades of an essay, we will plug the predictive model into the backend to evaluating essays from the user. Based on how good our accuracies are and if time pertains, we can feed new essays into the dataset and create a user feedback tool.

The Data

Our dataset for training and testing our algorithms is diverse. We obtained the data from kaggle.com by the William and Flora Hewlett Foundation (**https://www.kaggle.com/c/asap-aes/data**). It consists of 8 sets of essays authored by students from grade 7 to 10. According to kaggle.com, each essay has distinct characteristics that should be used for grading. Using these distinct characteristics, our algorithm will be trained better and will provide smarter evaluations. Each essay has human grades assigned to them and thus our algorithms can be supervised. The essays in the dataset are around 150 to 550 words in length. More so, each essay set also has a different grading rubric which will help our algorithm in being more prepared for general cases.

To solve the problem of over-fitting, we will randomly split the dataset into 2 parts:

1. Training Data - containing 80% of the dataset
2. Testing Data - containing 20% of the dataset

As we start evaluating our classifiers based on their success metrics after training, we may modify our datasets during the sprints.

To be able to evaluate an essay in the training data set, we first need to transform it into a feature vector. This vector can then be an input to the classifiers. We will transform essays into a feature vectors using python's Word2vec library. These feature vectors can be interpreted by the classifiers.

Django Application Server and Web Client

The browser based front end of the system will be built on python's efficient Django framework. The management of routes will be done on the application server to handle requests and to send back data. The front end will be implemented in HTML ,CSS, Javascript and the Django template language. We will use Angular libraries to simplify programming and frameworks like Bootstrap to design a better UI. The bulk of the content will be sent to the client in response to standard HTTP requests from the application server. If time permits, we may add a "Sign in with facebook” option for our users. To keep things simple and concentrate on our algorithm accuracy, we created 3 initial pages for a user to interact with our system. The UI mock ups for these pages can be viewed in later sections of the design document.

Firebase DB

As we mentioned earlier, the database will contain user information and past essay data. The application server module will make calls to this database to authenticate users and retrieve past essay data. To facilitate communication between the two modules we will use a python wrapper called Pyrebase. Using Firebase DB, our users will be able to access their data in realtime from any device using their browser. This database is also free for users with a small storage need. This will let us test our tool before we open it for many users. Firebase DB integrates with Firebase Authentication to provide simple and intuitive authentication for our users. Keeping the user’s data private is important for their trust. Firebase DB is secure, scalable and efficient. It will also seamlessly integrate with our django modules.

Class Design

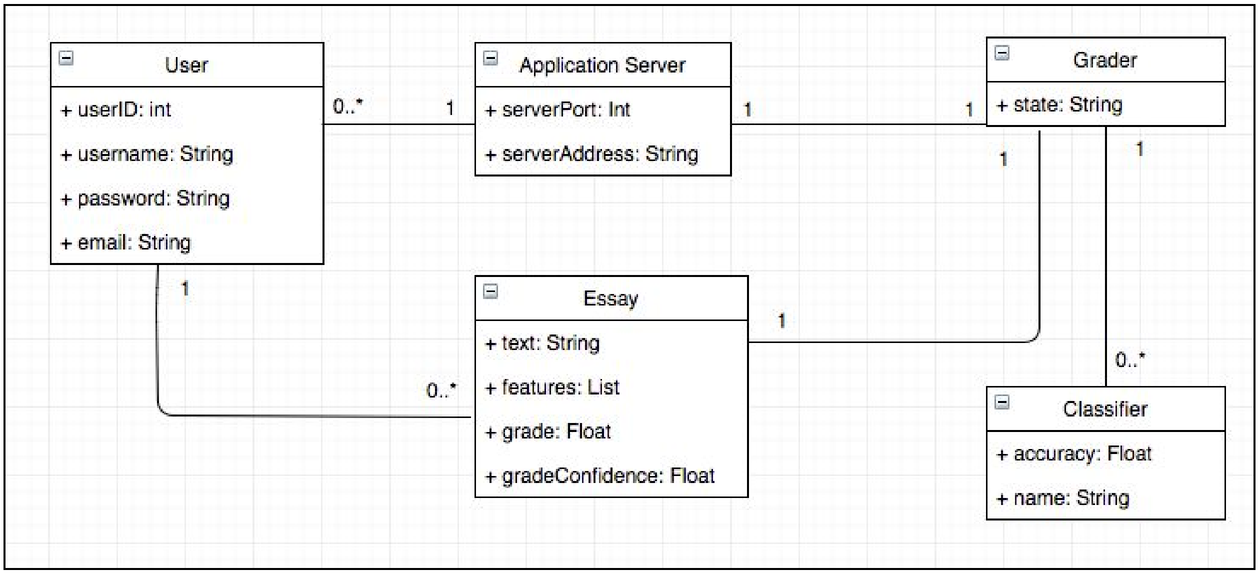


Figure 3: Class diagram depicted in terms of entity-relationships

User

The User class represents everything specific to a user such as the userId, password and email. This information will be specific to each user object that interacts with our system. The user is uniquely identified using the userID and is authenticated using their own created password (during sign up). Each user can have 0 or many essays that he or she has graded using our tool. The user can create more essays by interacting with the application server.

Application Server

The Application Server class represents the service class that holds everything specific to the web client such as server port and server address. The user interacts with this class to grade and access essays. Operations carried out by this class would be to display data to the user, to handle HTTP requests and to interact with the Grader class.

Grader

The Grader class represents the predictive machine learnt model of our system and takes essays as inputs and returns their grade after evaluating them. This class will have access to classifier objects and will be responsible to train and test these classifiers. It will take essays as inputs and generate feature vectors of those essays. These vectors will be sent to the classifier objects to retrieve grades from them.

Essay

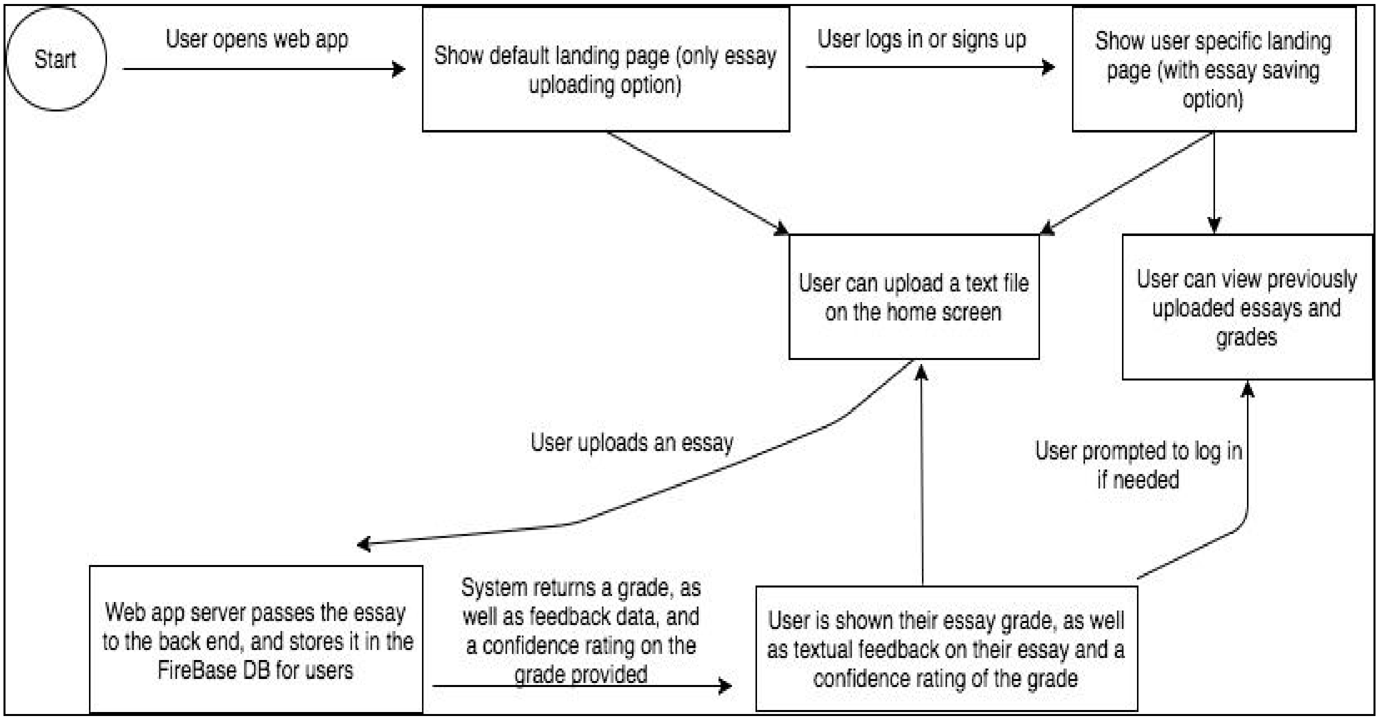
The Essay class represents essays with the essay text, features, grade and confidence scores. The user only enters the essay text and then the Grader evaluates it and generates a feature vector. This vector is given to the classifiers by the Grader which then generates a predicted score. The Grader evaluates the predicted scores from different classifiers and then assigns a confidence score to the essay.

Classifier

The Classifier class represents each machine learning algorithm in our tool. Each classifier has different accuracies and the Grader can use these to weight the results of different classifiers. The Grader is responsible for initializing new classifiers and then training and testing them.Multiple classifiers can be used to provide higher accuracies during predictions.

Activity Diagram

The activity diagram describes the user interaction with the system. It also shows the flow of different user activities like sign in, sign up, view past essays, upload essays and view grades.



Start
User opens web app
Show default landing page (only essay uploading option)
User logs in or signs up
Show user specific landing page (with essay saving option)
User can upload a text file on the home screen
User can view previously uploaded essays and grades
User uploads an essay
User prompted to log in if needed
Web app server passes the essay to the back end, and stores it in the FireBase DB for users
System returns a grade, as well as feedback data, and a confidence rating on the grade provided
User is shown their essay grade, as well as textual feedback on their essay and a confidence rating of the grade

Figure 4: Activity Diagram

Sequence Diagrams

Sign up and Sign in

Figures 5 and 6 show the sequence diagram of the user login and signup process. It shows the tasks carried out by the client(user), server and database.

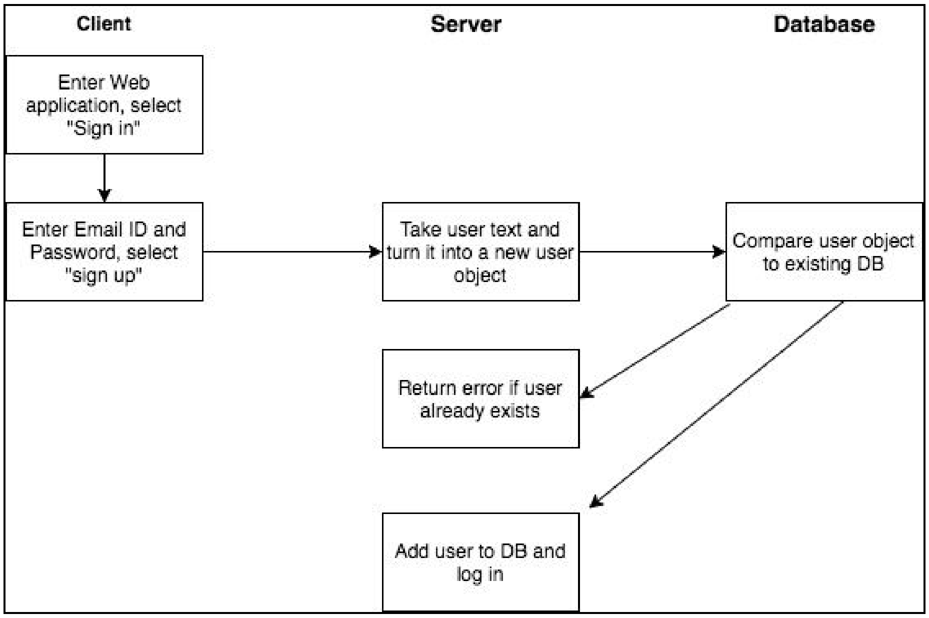

Client
Server
Database
Enter Web application, select "Sign in"
Enter Email ID and Password, select "sign up"
Take user text and turn it into a new user object
Compare user object to existing DB
Return error if user already exists
Add user to DB and log in

Figure 5: Sequence Diagrams for sign up

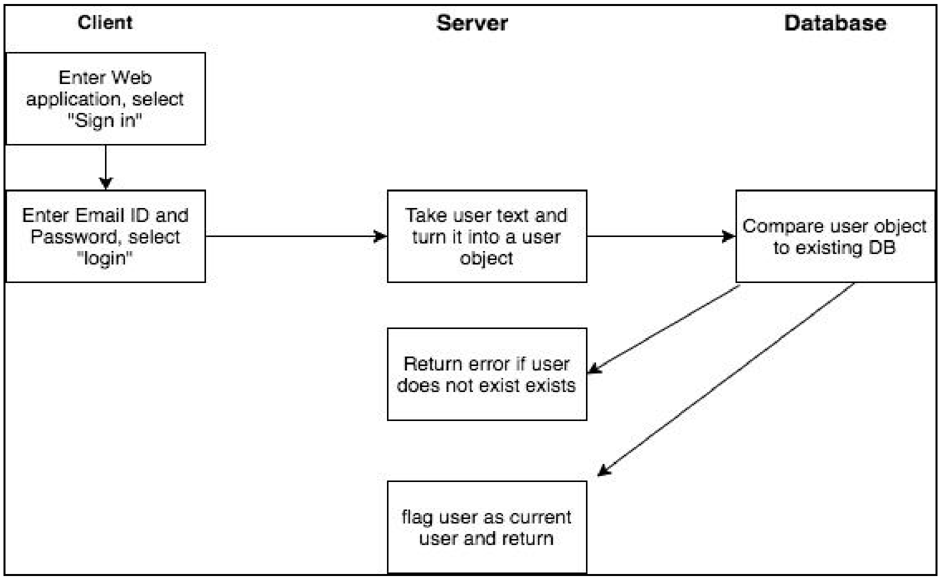

Client
Server
Database
Enter Web application, select "Sign in"
Enter Email ID and Password, select "login"
Take user text and turn it into a user object
Compare user object to existing DB
Return error if user does not exist exists
flag user as current user and return

Figure 6: Sequence Diagrams for sign in

View Past Essays

Figure 7 shows the sequence diagrams for viewing saved essays:

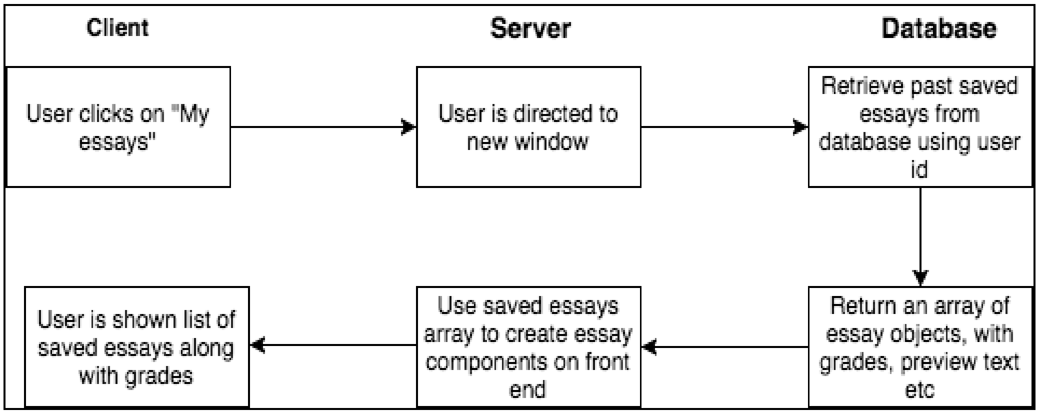

Client
Server
Database
User clicks on "My essays"
User is directed to new window
Retrieve past saved essays from database using user id
Return an array of essay objects, with grades, preview text etc
Use saved essays array to create essay components on front end
User is shown list of saved essays along with grades

Figure 7: Sequence Diagrams for viewing saved essays

Upload Essay

Figure 8 shows the sequence diagrams for uploading an essay and viewing its grade:

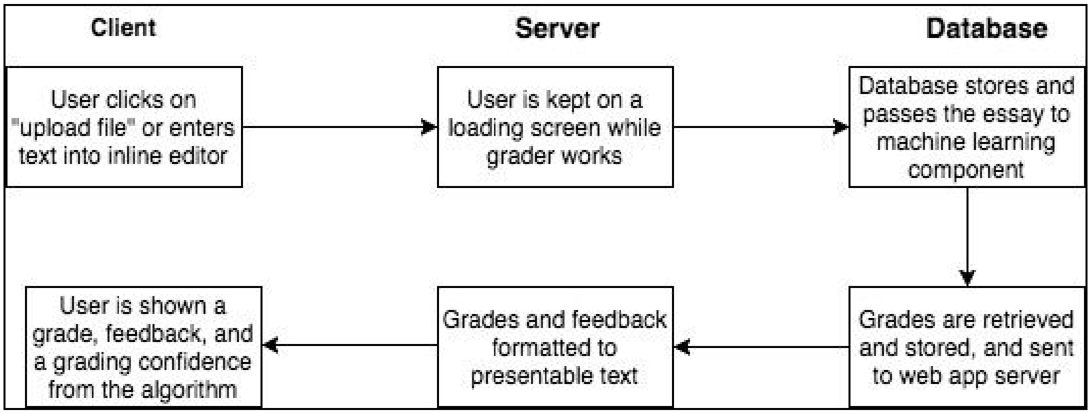

Client
Server
Database
User clicks on "upload file" or enters text into inline editor
User is kept on a loading screen while grader works
Database stores and passes the essay to machine learning component
Grades are retrieved and stored, and sent to web app server
Grades and feedback formatted to presentable text
User is shown a grade, feedback, and a grading confidence from the algorithm

Figure 8: Sequence Diagrams for getting an essay graded

User Interface

We took steps to create a simple yet intuitive user interface. Since the most important aspect of our project is the accuracy of the essay grades, we only have 3 initial pages that the user can interact with. These interfaces will enable a user to sign in, sign up, view past essays, upload essays and view grades.

The signup and login UI mock up is shown below (Figure 9):

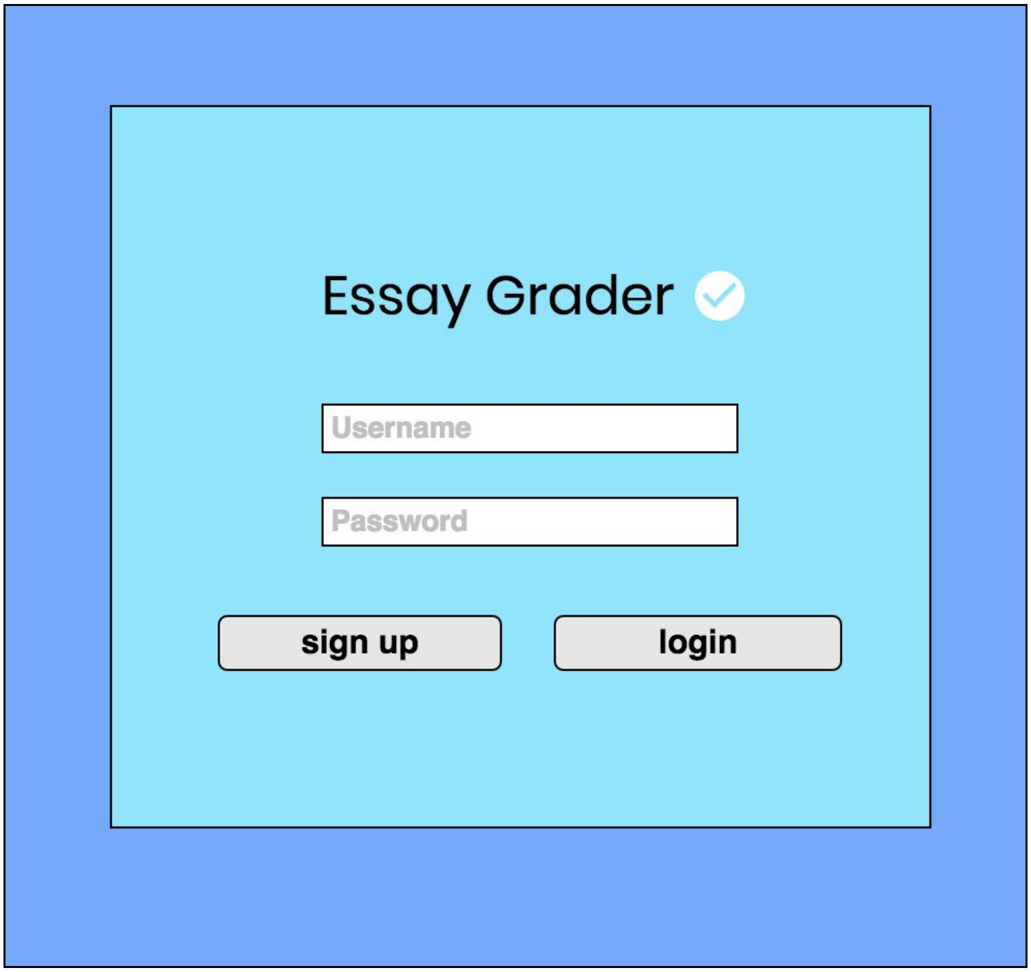
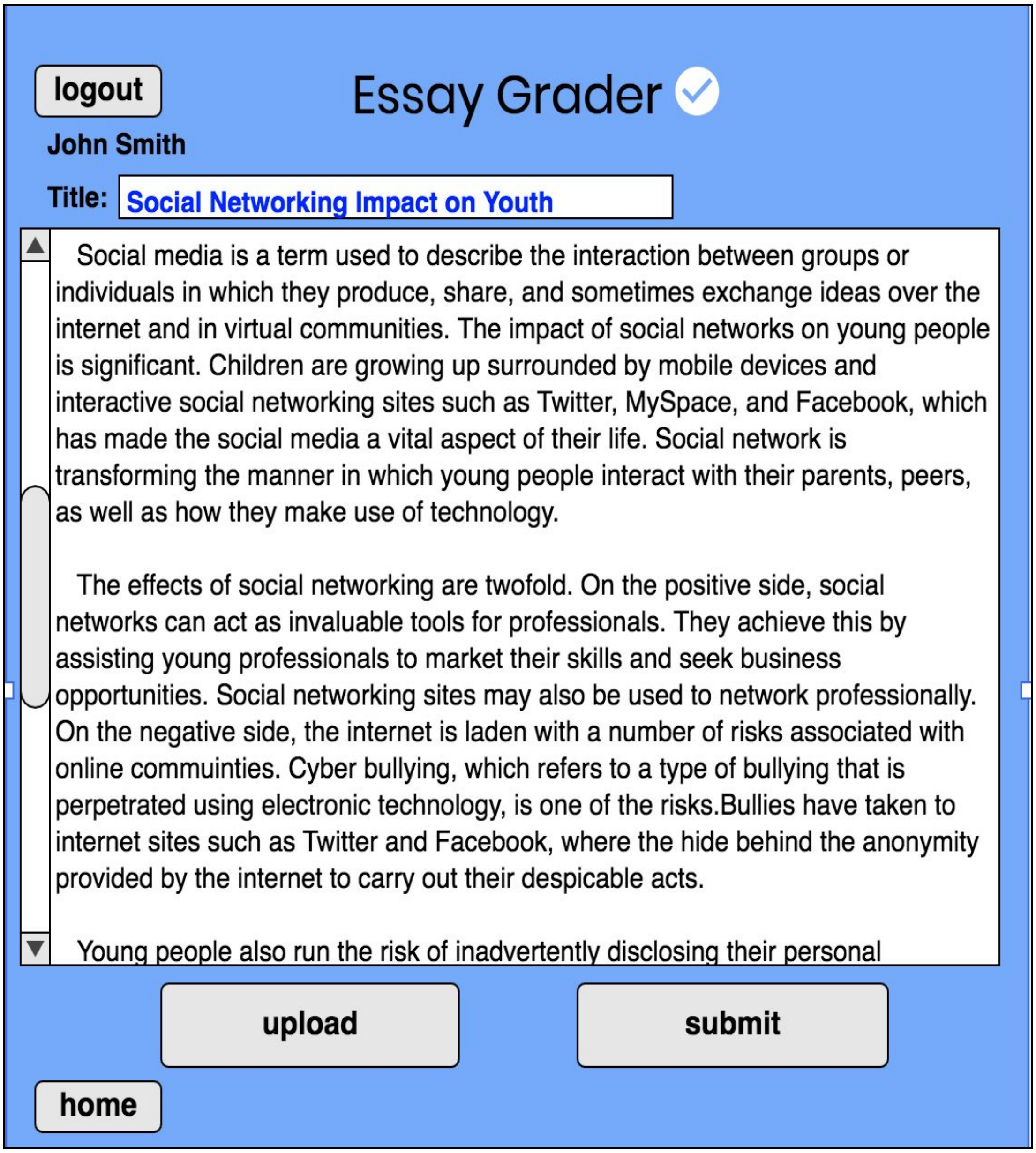


Figure 9: UI mock up for login and sign up

The upload page's mock up is shown below (Figure 10):



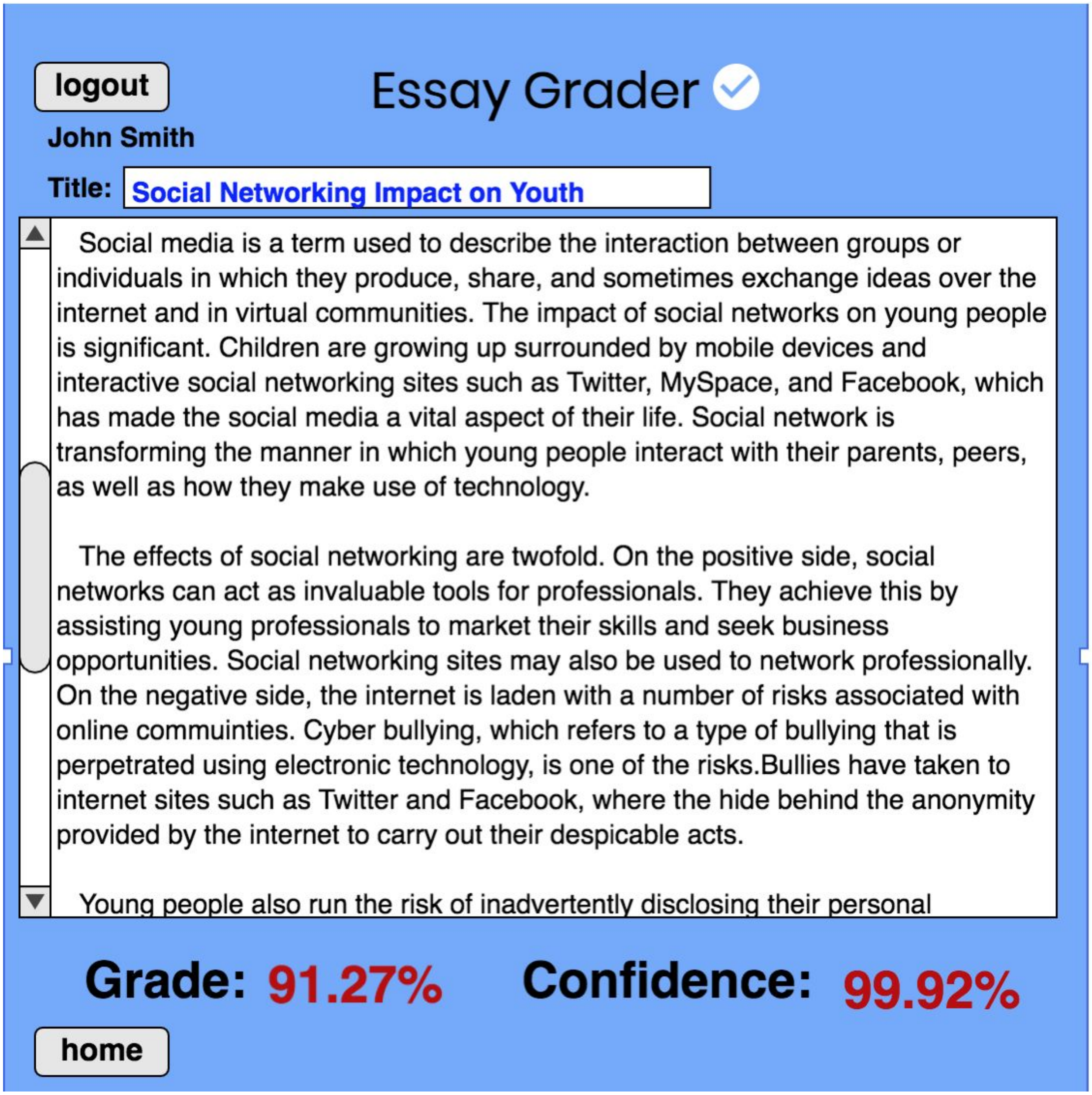
Social media is a term used to describe the interaction between groups or individuals in which they produce, share, and sometimes exchange ideas over the internet and in virtual communities. The impact of social networks on young people is significant. Children are growing up surrounded by mobile devices and interactive social networking sites such as Twitter, MySpace, and Facebook, which has made the social media a vital aspect of their life. Social network is transforming the manner in which young people interact with their parents, peers, as well as how they make use of technology.

The effects of social networking are twofold. On the positive side, social networks can act as invaluable tools for professionals. They achieve this by assisting young professionals to market their skills and seek business opportunities. Social networking sites may also be used to network professionally. On the negative side, the internet is laden with a number of risks associated with online commuinties. Cyber bullying, which refers to a type of bullying that is perpetrated using electronic technology, is one of the risks.Bullies have taken to internet sites such as Twitter and Facebook, where the hide behind the anonymity provided by the internet to carry out their despicable acts.

Young people also run the risk of inadvertently disclosing their personal

Figure 10: Upload Essay UI mock up

The grade-result page mock up is shown below (Figure 11):



John Smith

Title: Social Networking Impact on Youth

Social media is a term used to describe the interaction between groups or individuals in which they produce, share, and sometimes exchange ideas over the internet and in virtual communities. The impact of social networks on young people is significant. Children are growing up surrounded by mobile devices and interactive social networking sites such as Twitter, MySpace, and Facebook, which has made the social media a vital aspect of their life. Social network is transforming the manner in which young people interact with their parents, peers, as well as how they make use of technology.

The effects of social networking are twofold. On the positive side, social networks can act as invaluable tools for professionals. They achieve this by assisting young professionals to market their skills and seek business opportunities. Social networking sites may also be used to network professionally. On the negative side, the internet is laden with a number of risks associated with online commuinties. Cyber bullying, which refers to a type of bullying that is perpetrated using electronic technology, is one of the risks.Bullies have taken to internet sites such as Twitter and Facebook, where the hide behind the anonymity provided by the internet to carry out their despicable acts.

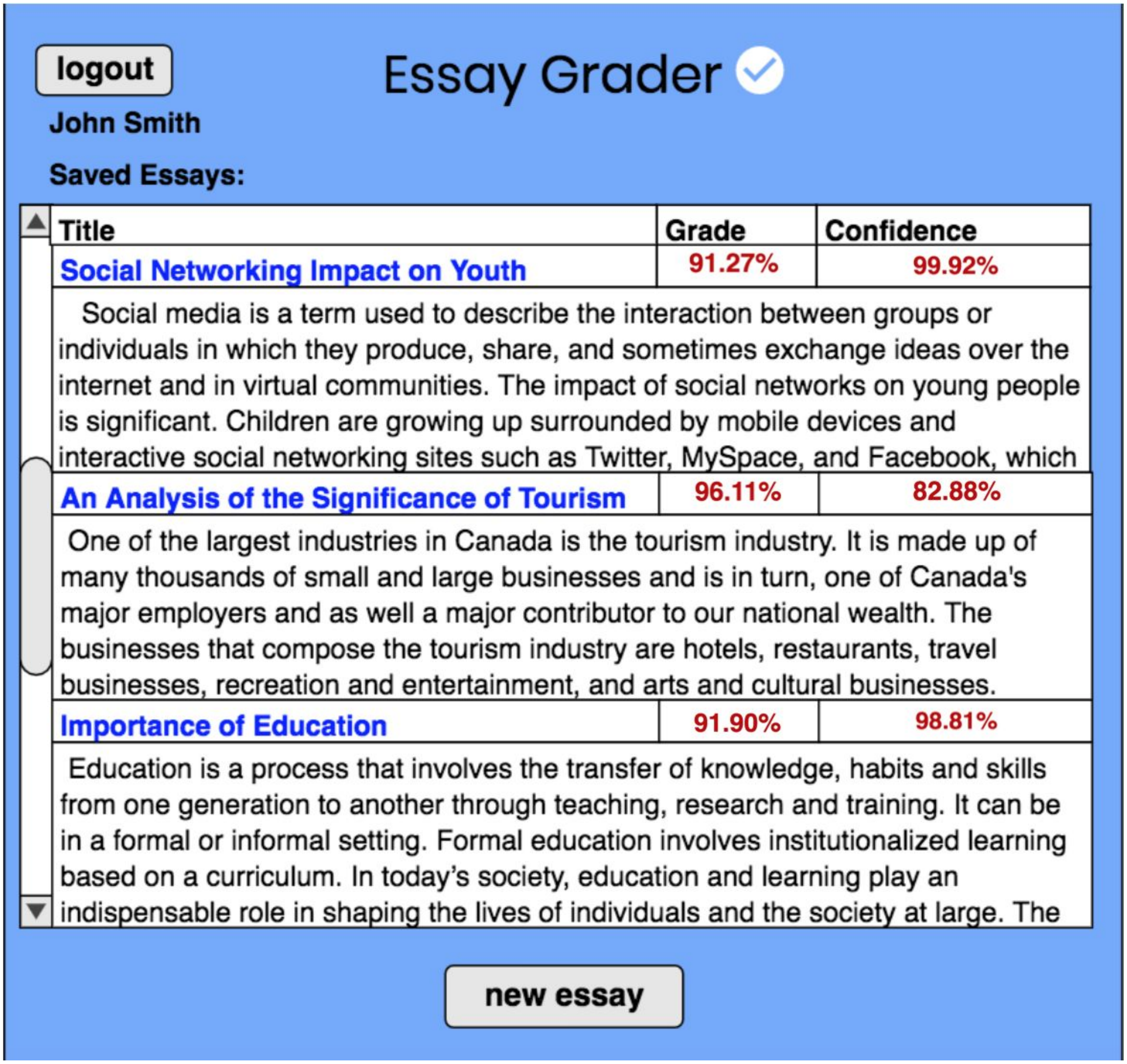
Young people also run the risk of inadvertently disclosing their personal

Grade: 91.27% Confidence: 99.92%

Figure 11: Result page UI mock up

The user can view the grade and the confidence of the algorithms on the same page as the essays. This way users can themselves read the essay and evaluate if the grade is accurate or not (This will help them get confidence on our tool !!).

The past essays page mock up is shown below (Figure 12):



John Smith

Saved Essays:

Title	Grade	Confidence
Social Networking Impact on Youth	91.27%	99.92%
Social media is a term used to describe the interaction between groups or individuals in which they produce, share, and sometimes exchange ideas over the internet and in virtual communities. The impact of social networks on young people is significant. Children are growing up surrounded by mobile devices and interactive social networking sites such as Twitter, MySpace, and Facebook, which		
An Analysis of the Significance of Tourism	96.11%	82.88%
One of the largest industries in Canada is the tourism industry. It is made up of many thousands of small and large businesses and is in turn, one of Canada's major employers and as well a major contributor to our national wealth. The businesses that compose the tourism industry are hotels, restaurants, travel businesses, recreation and entertainment, and arts and cultural businesses.		
Importance of Education	91.90%	98.81%
Education is a process that involves the transfer of knowledge, habits and skills from one generation to another through teaching, research and training. It can be in a formal or informal setting. Formal education involves institutionalized learning based on a curriculum. In today's society, education and learning play an indispensable role in shaping the lives of individuals and the society at large. The		

Figure 12: Past essay page mock up