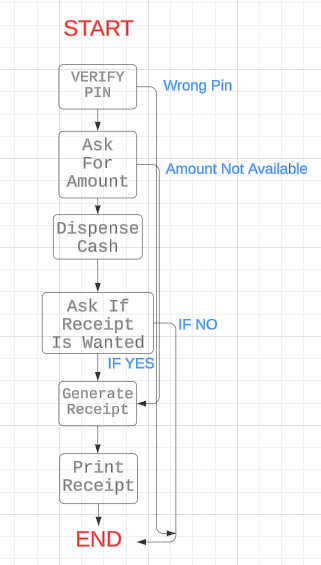
### **CS 255 System Design Document Template**

### 

### **UML Diagrams**

### 

### **UML Use Case Diagram**



**UML Activity Diagrams**

### 

### 

### 

### **UML Sequence Diagram**

### 

### 

### 

### 

### 

### 

### 

### 

### **UML Class Diagram**

### 

### 

### **Technical Requirements**

**Hardware Requirements:**

Client Side:

* Smartphone, tablet, or desktop/laptop w/ access to the internet
* Stable Internet connection

Server Side:

* Cloud Hosting platform like Google Cloud, or Microsoft Azure in order to host the app and database
* The System will need a minimum of 1TB of SSD Storage to manage data, training material, and logs.

**Software Requirements:**

Client Side:

* We will need typical front end development languages like JavaScript, HTML, and CSS. Frameworks like React.js will also be required to keep the user interface fast and dynamic.
* For mobile development the use of a cross platform framework like React Native will be needed for mobile apps.

Server Side:

* Backend development will need languages such as Java, Python, or Node.js.
* We will need NGINX or Apache for handling HTTP requests
* The use of databases like MySQL and NoSQL will be needed for both a relational and non-relational database.
* We will be using JWT for role-based-access control
* Finally having API integration for DMV updates, and payment gateways is required.

**Development Tools:**

* We will use IDEs such as VSCode, and PyCharm.
* Version Control: Git through GitHub
* Cypress for UI testing, Postman for API testing, and PyTest for backend testing
* We will use Azure DevOps for automated deployment and testing

### 

### 

### **Infrastructure Requirements:**

Hosting & Deployment:

* We will need Cloud services for hosting, such as Google App Engine
  + We will support this by using CDN like Cloudflare for reduced latency and improved performance

Security:

* Data encryption: TLS/SSL for secure data transfers
* Database: Encrypted storage for user data and backups
* Brute Force Protection: Account lock mechanisms will be implemented to combat multiple failed login attempts
* Monitoring: We will leverage IDS to log suspicious activity

Scalability:

* Auto-scaling in our cloud infrastructure to handle user usage spikes
* We will leverage loud balancers to distribute work during heavy traffic
* Built on a modular architecture