



**VCU** College of Engineering

# 304 Capital One Incident Wizard Design Report

Prepared for  
Capital One Vinay Soni  
Capital One

By  
Isaac Lyu, Gamal Almareh, Matthew Baker, Ahmad Scruggs

Under the supervision of  
Tomasz Arodz

Date 10/8/24

## **Executive Summary**

Capital One agents face growing challenges in managing a high volume of customer incidents, leading to delayed resolutions and reduced customer satisfaction. To address this, a Chat AI bot has been implemented to streamline the incident-handling process. This AI Chatbot is designed to increase the efficiency of agents not only in submitting incidents but also in ensuring null or unnecessary incidents are not sent into ServiceNow.

In this project we are creating an AI Tool that streamlines the process of creating customer incidents onto ServiceNow. Currently, Capital One agents are required to manually transfer data from their Capital One Environment onto the third-party service called ServiceNow. ServiceNow is a cloud-based platform that provides IT service requests and customer service management. Capital One agents use ServiceNow to create “incidents” which are issues that cannot be solved by the agent. The agent must manually fill in all the necessary data for the incident so that it is all transferred onto the third-party platform. This transfer of information loses a lot of the contextual information that could help in the solution of the incident therefore causing most incidents to be ignored or deemed unsolvable by the developer processing the incident request. This is where the AI Incident Tool comes in. This tool would allow agents to easily and efficiently transfer ALL data onto the ServiceNow Platform. This would allow the agent to provide higher quality customer service, allow less garbage incidents to be put into ServiceNow, and allow engineers to spend more time on incidents that are actually able to be fulfilled due to the extra contextual data.

## Table of Contents

Section A. Problem Statement	5
Section B. Engineering Design Requirements	7
B.1 Project Goals (i.e. Client Needs)	7
B.2 Design Objectives	7
B.3 Design Specifications and Constraints	8
B.4 Codes and Standards	9
Section C. Scope of Work	11
C.1 Deliverables	11
C.2 Milestones	12
C.3 Resources	12
Section D. Concept Generation	13
Section E. Concept Evaluation and Selection	14
Section F. Design Methodology	16
F.1 Computational Methods (e.g. FEA or CFD Modeling, example sub-section)	16
F.2 Experimental Methods (example subsection)	16
F.5 Validation Procedure	16
Section G. Results and Design Details	18
G.1 Modeling Results (example subsection)	18
G.2 Experimental Results (example subsection)	18
G.3 Prototyping and Testing Results (example subsection)	18
G.4. Final Design Details/Specifications (example subsection)	18
Section H. Societal Impacts of Design	20
H.1 Public Health, Safety, and Welfare	20
H.2 Societal Impacts	20
H.3 Political/Regulatory Impacts	20
H.4. Economic Impacts	20
H.5 Environmental Impacts	21
H.6 Global Impacts	21

H.7. Ethical Considerations	21
Section I. Cost Analysis	22
Section J. Conclusions and Recommendations	23
Appendix 1: Project Timeline	24
Appendix 2: Team Contract (i.e. Team Organization)	25
Appendix 3: [Insert Appendix Title]	26
References	27

## **Section A. Problem Statement**

The problem that we are addressing is that creating ServiceNow incidents is a tedious process that affects an agent's efficiency. By cutting out some steps using the AI Chatbot we can effectively increase productivity, and quality of customer service.

**Some problems that we want to solve with our AI Tool are as follows:**

- How can we increase the efficiency of agents using the AI Chatbot?
- How can we improve quality of customer service using the AI Chatbot?
- In what ways can we make the AI interactive with the agent?
- How can we increase the efficiency of the engineers accessing the incidents on ServiceNow?

**Unmet Engineering Needs:**

- There isn't a system for creating incidents for ServiceNow. Currently it is all manual with the agent.

**Who is affected by the problem at hand?**

The issues that we are trying to find solutions for mainly affect the agents, the customers and the engineers. Because of the lack of an efficient system/tool to streamline the process, the agents and engineers are bogged down by the manual process of creating and troubleshooting the incident leading to lower efficiency and quality in turn affecting the customer's satisfaction.

**Field of Study and Industry Context:**

This project falls under the field of artificial intelligence (AI) and its application in the customer support and IT service management (ITSM) industries. The use of AI-powered chatbots to assist in routine tasks, such as troubleshooting and incident management, has gained prominence in recent years, driven by the growing need for efficiency and automation in large organizations. Chatbots can reduce the time spent on repetitive tasks, improve customer satisfaction, and enable support agents to focus on more complex problems.

In the context of ITSM, platforms like ServiceNow are critical for managing incidents, service requests, and changes within organizations. However, the process of creating, categorizing, and submitting incidents often remains manual, leading to delays and inconsistencies. This project aims to improve these workflows by integrating an AI-driven chatbot into the incident management process used by Capital One agents.

**Sponsor Company and Relevance:**

Capital One, a major financial institution, relies heavily on efficient IT operations to ensure seamless service delivery to its customers. The project is aligned with Capital One's broader objective of leveraging AI to optimize internal workflows. The chatbot will interface with

ServiceNow, a widely used ITSM platform, to assist agents in solving issues, reducing human error, and ensuring that incidents are categorized and submitted accurately.

### **Historical Perspective and Previous Solutions:**

The application of AI in ITSM is not a new concept. Companies such as IBM have developed AI systems like Watson that assist in problem diagnosis and resolution. These systems have been successful in reducing response times and improving ticket resolution. Similarly, AI chatbots such as IPsoft's Amelia have been used to handle customer service interactions, demonstrating the feasibility and benefits of integrating AI into service management.

However, while these systems have seen success, they often lack customization for specific workflows or industries, or they focus more on customer-facing applications rather than internal IT agent support. The proposed chatbot aims to address these gaps by providing a solution specifically tailored to Capital One's incident management process. By combining conversational AI with domain-specific knowledge, the chatbot will assist agents in real time, potentially reducing the time spent on incident resolution and improving data consistency across the board.

### **Competitive Landscape and Prior Solutions:**

There are several commercially available solutions that tackle IT service management automation, such as ServiceNow's own virtual agent, which helps users navigate the platform and create incidents. However, this virtual agent is often generic and does not provide deep insights into the context of a company's unique IT structure.

This project proposes an alternative design that will build upon existing technologies but with a focus on customization for Capital One's needs. By incorporating advanced natural language processing (NLP) and machine learning models, this AI chatbot will improve the incident submission process and ensure that agents are better supported in their tasks.

### **Improvements Over Previous Solutions:**

Existing chatbots and AI systems in ITSM generally automate the submission process but do not offer robust troubleshooting capabilities that would help agents solve issues prior to submission. This project's chatbot not only aims to automate the creation of incidents but also provides relevant suggestions based on historical incidents and solutions, enabling agents to resolve incidents faster and more accurately.

This tailored, interactive experience ensures that Capital One agents can address incidents more efficiently, ultimately contributing to greater productivity and reduced downtime.

## Section B. Engineering Design Requirements

This section describes the goals and objectives of the AI Incident Wizard project, outlines the design specifications and constraints, and identifies the relevant code and standards that will guide the development process. The design requirements are defined from the needs of Capital One's customer service operations, with a focus on improving incident management and enhancing overall service quality.

### B.1 Project Goals (i.e. Client Needs)

The primary goal of this project is to enhance the efficiency and quality of customer service at Capital One through the implementation of an AI incident Wizard. By addressing the current challenges faced by agents, the project aims to streamline incident management and reduce unnecessary workload. The specific goals include:

- **Improve Agent Efficiency:** Enables agents to focus on customer interactions without the distraction of manually creating incidents.
- **Reduce Unnecessary Data:** Minimize the creation of voided incidents to optimize data storage and management.
- **Enhance Incident Resolution:** Provide agents with AI-driven solutions to resolve incidents promptly, improving overall customer satisfaction.
- **Utilize agent feedback :** Presents incidents to agents before transferring unresolved incidents to ServiceNow, ensuring timely follow-up and resolution

### B.2 Design Objectives

The design of the AI incident Wizard will be guided by key objectives that outline what the system will accomplish. These objectives will be SMART- Specific, Measurable, Achievable, Realistic, and Time-bound. The design objectives include:

- **The design will enable agents to submit incidents through an intuitive interface, ensuring ease of use and accessibility.**
- **The design will integrate AI capabilities to facilitate quick decision making.**
- **The design will include a feedback mechanism that allows agents to rate AI suggestions ensuring continuous improvement and quality.**
- **The design will be developed in a timely manner that allows for iterative testing and adjustments based on user feedback.**

### **B.3 Design Specifications and Constraints**

- The design should respond to user input quickly (within 2-3 seconds)
- The design should integrate with ServiceNow's API
- The design should be able to handle at least 20 concurrent users
- The design should be able to maintain a 95%+ uptime
- The design should have username and password authentication at least
- The design should store data in a secure fashion using ServiceNow as a database
- The design should be within the given budget of \$1000
- The design should streamline the incident creation process
- The design should allow incidents to be stored in ServiceNow without the agent leaving the service application they are already in

### **B.4 Codes and Standards**

- OWASP Standards – design must validate user input and provide basic encryption
- Web Development Standards – design must work well across many browsers and screen sizes
- Software Testing Standards – design must include unit tests to validate key functions



## Section C. Scope of Work

The project scope defines the boundaries of the project encompassing the key objectives, timeline, milestones and deliverables. It clearly defines the responsibility of the team and the process by which the proposed work will be verified and approved. A clear scope helps to facilitate understanding of the project, reduce ambiguities and risk, and manage expectations. In addition to stating the responsibilities of the team, it should also explicitly state those tasks which fall *outside* of the team's responsibilities. *Explicit bounds* on the project timeline, available funds, and promised deliverables should be clearly stated. These boundaries help to avoid *scope creep*, or changes to the scope of the project without any control. This section also defines the project approach, the development methodology used in developing the solution, such as waterfall or agile (shall be chosen in concert with the faculty advisor and/or project sponsor). Good communication with the project sponsor and faculty advisor is the most effective way to stay within scope and make sure all objectives and deliverables are met on time and on budget.

### C.1 Deliverables

- Project Proposal - outlines the objectives, scope, methodology, and resource requirements for the Capital One Incident Wizard Program, serving as a blueprint to guide development and secure stakeholder approval.
- Fall Design Poster- visually summarizes the key elements of the Incident Wizard Program, including project goals, design concepts, and implementation strategies for presentation at events.
- Preliminary Design Report- details the initial design concepts, technical specifications, and chosen technologies for the Incident Wizard Program, evaluating feasibility and outlining next steps for development.
- User Interface Prototype: A functional front-end built with React, showcasing the incident management dashboard.
- Backend API: A Flask-based API for incident routing, data processing, and communication between the UI and internal systems.
- AI Model Integration: Implementation of OpenAI 4.0 for automated incident categorization and resolution suggestions.
- Workflow Automation: Integration of ServiceNow to automate ticketing, task assignment, and incident tracking.
- GitHub Repository: A GitHub repository with version control, including all source code, documentation, and setup instructions.
- Final Project Submission- fully developed solution, including the complete system implementation, documentation, and testing results.

## **C.2 Milestones**

- Project Proposal - Oct 11
- Design Poster - Nov 15
- Preliminary Design Report - Dec 9
- UI Prototype - Feb 26
- Backend API - Mar 19
- AI Model Integration & Workflow Automation -Apr 30
- Github Repo & Final Project Submission - May 3

## **C.3 Resources**

- React - Enables a responsive and interactive UI, allowing users to track and manage incidents in real-time with efficient, reusable components.
- Flask - Provides a simple yet powerful backend for routing incidents, processing API requests, and integrating with internal systems for smooth operations.
- OpenAI 4.0- Automates incident categorization and prioritization using AI, offering intelligent resolution suggestions to reduce manual work and speed up the process.
- ServiceNow - Streamlines workflows by automating ticket creation, task assignments, and incident tracking, ensuring efficient management from start to finish.
- Github- Facilitates version control, collaboration, and continuous deployment, allowing teams to develop, review, and update the program without disrupting operations.

## Section D. Concept Generation

### Concept Design: AI-Powered Incident Wizard

#### Core Components

##### 1. Intelligent Incident Creation Interface

- A sleek, intuitive web-based interface that guides users through the incident creation process.
- Responsive design adapting to various screen sizes and browsers for seamless access across devices.

##### 2. AI-Driven Data Compilation

- An advanced AI engine that:
  - Analyzes user input in real-time
  - Suggests relevant fields and information based on the incident type
  - Learns from historical data to improve accuracy over time

##### 3. Context-Aware Chatbot

- An AI-powered chatbot that:
  - Engages users in natural language conversations
  - Asks targeted questions to gather comprehensive incident details
  - Provides instant suggestions and clarifications

##### 4. Dynamic Feedback System

- A built-in rating mechanism allowing agents to evaluate AI suggestions
- Machine learning algorithms that continuously refine the AI based on this feedback

##### 5. Secure Authentication Portal

- A robust login system with multi-factor authentication options
- Role-based access control to ensure data security and privacy

#### User Experience Flow

1. Login: Users securely access the system through the authentication portal.
2. Incident Initiation: Upon login, users are greeted by the AI chatbot, which initiates the incident creation process.
3. Guided Information Gathering: The chatbot engages in a conversational flow, intelligently asking for relevant details based on the incident type and user responses.

4. AI-Assisted Form Filling: As the conversation progresses, the AI automatically populates the incident form fields, with the option for users to review and edit.
5. Context Enhancement: The AI suggests additional context based on similar past incidents, helping to provide a more comprehensive picture for engineers.
6. Review and Submit: Users review the compiled incident report, make any necessary adjustments, and submit.
7. Feedback Loop: After submission, users can rate the AI's performance, contributing to its ongoing improvement.

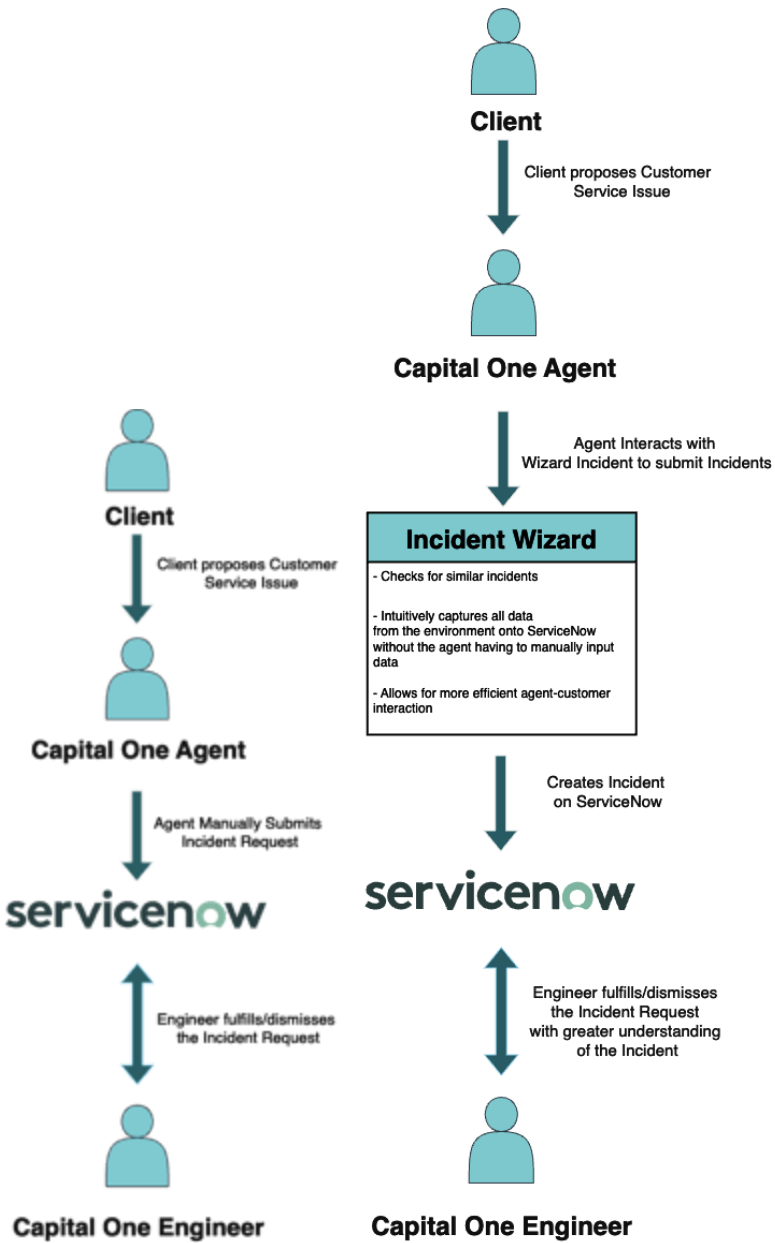
## Visual Design Elements

- A clean, modern interface with a color scheme aligned with ServiceNow's branding
- Intuitive icons and visual cues to guide users through the process
- Progress indicators showing the stages of incident creation
- Collapsible sections for easy navigation through longer forms
- Real-time validation indicators to ensure data completeness and accuracy

## Backend Integration

- Seamless integration with ServiceNow's existing infrastructure
- Real-time synchronization with the ServiceNow database
- API endpoints for potential future integrations with other systems

By combining these elements, the AI-Powered Incident Wizard concept aims to significantly reduce incident creation time, improve data quality, and provide engineers with the context they need to resolve issues efficiently. This innovative approach promises to enhance both agent productivity and customer satisfaction in the incident management process. These images below depict how the process would normally work and with our implementation how it would be improved.



## **Section E. Concept Evaluation and Selection**

### **Evaluation Criteria:**

The criteria that drives our decision making process for our AI-Chatbot are ease of use, efficiency, and functionality. We want to make sure that any design choice we make we are ensuring that those three criteria are met. To ensure this we are implementing ways for our AI-Chatbot to be able to be implemented seamlessly into any environment, be easy to use, and be able to fulfill its function of sending incidents to ServiceNow with the appropriate context and information necessary.

### **General Alternatives:**

#### **Design Concept 1:**

At first we were considering creating a design that was specifically centered around being run on the Capital One framework and environment using the same tools as Capital One since we were doing this project sponsored by them. This design was deemed too difficult to implement because it would require us as students to have access to the tools at Capital One which wasn't possible. Our next issue was that since we don't work at Capital One and don't have the necessary contextual knowledge to understand the incidents specific to Capital One, we wouldn't be able to optimally implement our chatbot to handle said issues. It also would not fulfill the criteria stated before as it would be too niche to be implemented in other use cases.

#### **Design Concept 2:**

For our second design, we opted to instead look into using tools that weren't part of the Capital One Framework. This change would simplify a lot of our design since instead of having to learn and use their tools, we can use the tools we are comfortable with already and that we have access too. Along with making it easier on our end to develop, it would also make it easier for it to be used in other use cases outside of Capital One. This, however, doesn't solve the issue of not having the necessary contextual knowledge to create a chatbot for the specific use-case of Capital One. Since we don't understand the background of the wide range of incidents that could possibly come through, we wouldn't be able to optimize this Chatbot for Capital One which leads to our third and final design concept.

#### **Design Concept 3:**

Our final design concept is one that abstracts all the intricacies of Capital One and instead fulfills incidents in a void. This concept would require us to create a generic company with our own set of possible incidents and contextual knowledge. This way we can cater our chatbot to this environment without needing to have all the background information from Capital One. This concept is one that would show proof of concept and prove to the potential users of this tool that it can be implemented to any design or functionality. This tool can be catered to be used in any department within Capital One or to other companies or users outside of Capital One. We believe that this design concept fulfills all the criteria necessary for our project to be successful.

## **Baseline Evaluation:**

Our baseline for comparison with our tool would be what Capital One is currently using to fulfill incidents. The current method is to have whoever is reporting and issue, to then manually go into ServiceNow and create an incident with whatever contextual information they could scrounge up and submit. This would be compared to our ChatBot which would interact with the person reporting the issue and automatically submit the necessary contextual information needed to successfully fulfill the incident on ServiceNow. The main issue with the baseline method is that it lacks ease of use for the person reporting the incident and often the incident on ServiceNow lacks the necessary context for the engineer to be able to adequately solve it. With our chatbot we aim to challenge this issue by providing better ease of use for the user and provide more readily available contextual information about the incident straight to ServiceNow.

## **Section F. Design Methodology**

During our design process, we will adopt an iterative approach similar to those used in agile development cycles. Every two weeks, we will conduct structured evaluation sessions to assess progress against the design objectives and specifications. These sessions will involve testing the current iteration of our AI-Chatbot using simulated scenarios. Feedback from these evaluations will be used to improve system functionality, optimize performance, and ensure alignment with client requirements.

To validate the design, we will integrate computational modeling and data analysis tools, such as Python-based libraries for machine learning (e.g., TensorFlow or PyTorch) to monitor model performance metrics like accuracy and response time. Assumptions in the AI model, such as the relevance of generated responses, will be tested under controlled scenarios to ensure reliability.

We will use tools like Figma or other user interface prototyping software to evaluate the design's visual and functional aspects in order to make adjustments in the future.

Verification will focus on ensuring the Incident Wizard meets all predefined technical specifications, such as response latency thresholds and integration requirements. Validation will ensure the system addresses client needs by demonstrating its ability to create actionable and accurate incident reports under varied and complex conditions

### **F.5 Validation Procedure**

Throughout the project we will be meeting with and demonstrating the current iteration of our project to our point of contact to the sponsor, Vinay Soni. We will be using Vinay's input and our iterative process in order to ensure that our product meets the requirements of the client.



## Section G. Results and Design Details

### Architecture Overview

#### 1. Docker Containerization

- Use Docker to containerize the chatbot application, ensuring a consistent and portable environment across different deployment stages (development, staging, production)<sup>3</sup>.
- Docker simplifies the management of dependencies and ensures the chatbot has access to the specific versions of libraries it needs.

#### 2. OpenAI Integration

- Utilize OpenAI's GPT-3 or similar models for natural language processing and understanding. This involves setting up an OpenAI API key and configuring the API endpoints to interact with the chatbot<sup>23</sup>.
- Ensure the `.env` file includes necessary environment variables such as `OPENAI_API_KEY`, `OPENAI_API_BASE`, and `OPENAI_API_VERSION` to connect to the OpenAI API<sup>1</sup>.

#### 3. ServiceNow API Integration

- Integrate the chatbot with the ServiceNow API to submit incidents. This involves setting up API credentials and configuring the API endpoints to send incident data to ServiceNow.
- Define the necessary fields and data validation to ensure accurate and complete incident reports are submitted.

#### 4. React Native for Frontend

- Use React Native to build the chatbot's user interface, allowing for cross-platform compatibility (iOS and Android).
- Design a user-friendly UI that guides users through the incident reporting process, ensuring clear and concise interactions<sup>2</sup>.

### Detailed Design Components

#### User Interface (UI) and User Experience (UX)

- Interaction Methods: Design the UI to support both text and speech interactions. Ensure the chatbot interface is intuitive and easy to navigate.
- Conversational Flow: Implement a conversational flow that logically guides users through the incident reporting process. Use AI to understand user inputs and respond accordingly<sup>1</sup>.

## AI and NLP

- Intent Detection: Use OpenAI's GPT-3 or similar models to detect user intents accurately. This involves training the model on relevant datasets to handle incident-related queries.
- Context Retention: Ensure the chatbot can retain context for up to a certain period (e.g., 10 minutes) to maintain coherent and relevant responses throughout the conversation<sup>1</sup>.

## Backend and API Integrations

- Spring Boot or Similar Backend Framework: Use a backend framework like Spring Boot to handle API requests and interactions between the chatbot and ServiceNow. This involves setting up REST APIs to receive user inputs and send incident data to ServiceNow<sup>2</sup>.
- ServiceNow API Configuration: Configure the ServiceNow API to accept incident reports from the chatbot. Ensure all necessary fields are populated correctly and data validation is in place.

## Docker Configuration

- Dockerfile and Container Setup: Create a Dockerfile that includes all dependencies required for the chatbot, such as OpenAI API keys, ServiceNow API credentials, and any other necessary libraries.
- Environment Variables: Use environment variables in a `.env` file to configure the chatbot's settings, such as API keys and endpoints<sup>1</sup>.

## Example Configuration and Deployment

### Environment Variables

text

# .env file example

DOMAIN=yourdomain.com

OPENAI\_API\_TYPE=gpt-3

OPENAI\_API\_BASE=https://api.openai.com/v1

OPENAI\_API\_KEY=your\_openai\_api\_key

OPENAI\_API\_VERSION=2023-10-27

OPENAI\_DEPLOYMENT\_ID=your\_model\_name

SERVICENOW\_API\_URL=https://your-servicenow-instance.service-now.com/api/now/table/incident

SERVICENOW\_API\_USER=your\_servicenow\_username

SERVICENOW\_API\_PASSWORD=your\_servicenow\_password

## **Docker Run Command**

bash

```
docker run -p 8080:8080 --env-file .env chatbot_AI:latest
```

## **React Native Setup**

- Ensure the React Native project is set up to communicate with the backend API hosted in the Docker container.
- Use APIs to fetch and send data between the frontend and backend.

## **Testing and Evaluation**

- Usability Testing: Conduct usability testing to ensure the chatbot is user-friendly and effective in collecting incident reports.
- Performance Metrics: Define metrics such as incident submission rate, user satisfaction, and error rates to evaluate the chatbot's performance.
- Integration Testing: Test the integration with ServiceNow to ensure incident reports are submitted correctly and accurately.

## **Conclusion**

By integrating Docker for containerization, OpenAI for NLP, ServiceNow for incident management, and React Native for the frontend, you can create a robust and efficient AI chatbot that streamlines the incident reporting process. This approach ensures a consistent, scalable, and user-friendly solution that meets the design objectives and user needs.

## **Section H. Societal Impacts of Design**

### **H.1 Public Health, Safety, and Welfare**

- With our project there is likely no health, safety or welfare impact as it will not affect anyone not working directly with the product.

### **H.2 Societal Impacts**

- Our project is unlikely to have any impact on society at large.

### **H.3 Political/Regulatory Impacts**

- There is the question of privacy when using AI, how much of one's personal data is being used? In our design we plan on ensuring that we use no other data than what is necessary.

### **H.4. Economic Impacts**

- Our project will not have any foreseeable major economic impacts.

### **H.5 Environmental Impacts**

- Because our project includes the use of AI, there are various societal impacts that we must consider. The first of these would be the potential environmental impact of this technology as we will be using OpenAI's ChatGPT 4.0, which consumes large quantities of water in order to operate.

### **H.6 Global Impacts**

- Our project will not have any foreseeable major global impacts.

### **H.7. Ethical Considerations**

- AI is trained using various data sources, some of which could be biased, thus leading to biased responses from the AI. This will hopefully not be an issue with our product as it is simply an assistant for creating incidents but it is important to keep in mind regardless.

## Section I. Cost Analysis

The only cost consideration for our project is that of the OpenAI API key we will need to purchase. This key is billed based on the amount of tokens sent to the bot and thus there is no definite answer to how much the project will cost. Though we can get an estimate as we know that the GPT-4o API key costs \$2.5/1M tokens sent, \$1.25/1M tokens cached, and \$10/1M tokens outputted. From this we can gather that the cost of the project including testing will likely be within \$100 dollars. This falls well below the budget threshold of \$1000 provided by VCU.

## Section J. Conclusions and Recommendations

The **Capital One Incident Wizard** project aimed to fix the inefficiencies in how agents create and manage incidents in ServiceNow. We focused on improving productivity, reducing unnecessary work, and making the process more user-friendly.

### How We Got to the Final Design

1. We started by analyzing the current manual process, which highlighted problems such as missing context in incidents and delays in resolution.
2. Our early ideas relied heavily on Capital One's internal tools, but we quickly realized that wasn't practical for us as students. So, we pivoted to building a more general proof-of-concept tool that could still meet the goals of the project.
3. The final design uses:
  - OpenAI's GPT-4 to categorize incidents and suggest solutions.
  - React for a clean, responsive front end.
  - Flask for handling backend tasks and connecting to ServiceNow.
  - Docker to keep everything consistent and portable.

### Lessons and Challenges

- **Understanding the Problem:** Since we didn't have direct access to Capital One's specific workflows, we created our own simulated environment to prove the concept.
- **Choosing the Right Tools:** Using technologies we were familiar with, like React and Flask, made it easier to develop and focus on the core functionality.
- **Sticking to the Budget:** By carefully selecting tools and APIs, we kept costs well below the \$1000 budget while delivering a functional solution.

### What's Next

1. **Customizing the Tool:** Future teams could fine-tune the AI to better handle Capital One-specific incidents and even add support for multiple languages.
2. **Improving Security:** While we covered the basics, adding advanced features like fraud detection or compliance with financial regulations would take it further.

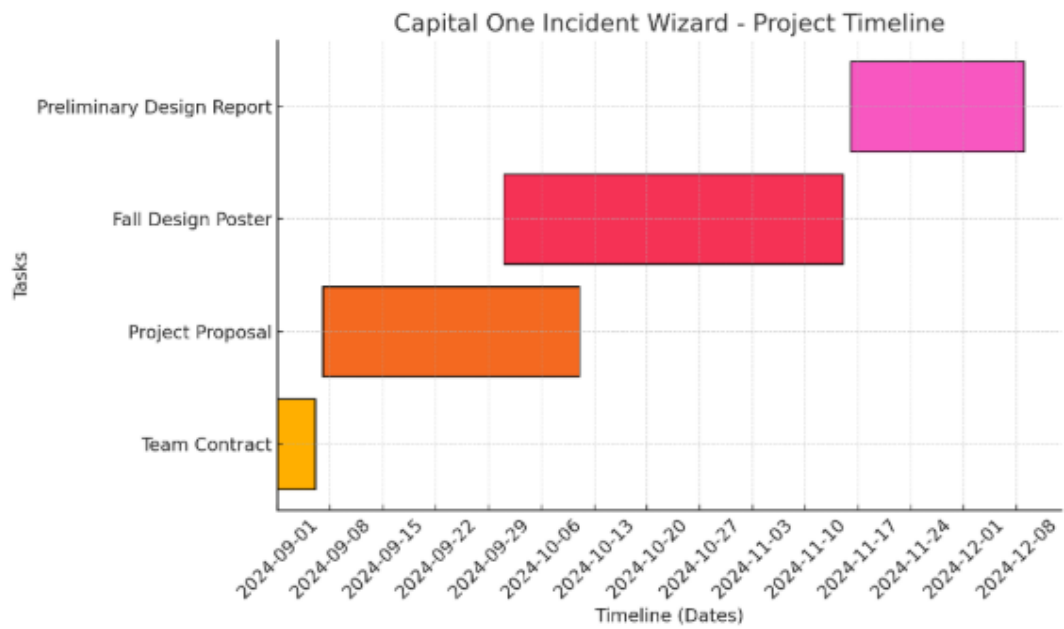
- 3. **Scaling Up:** The tool currently supports 20 users at a time, but future versions could optimize for larger teams and faster response times.
- 4. **Real-World Testing:** Deploying it in Capital One’s environment and gathering feedback from real users would be the ultimate test of its value.

**Final Thoughts**

The **Capital One Incident Wizard** is a solid starting point for making incident management faster and more reliable. We’ve shown how AI can streamline this process and save agents and engineers a lot of time. Moving forward, with more customization and testing, this tool could really transform how incidents are managed—not just at Capital One but in similar IT environments.

**Appendix 1: Project Timeline**

Provide a Gantt chart of similarly composed visual timeline showing the start and end dates of all completed tasks and how they are grouped together, overlapped, and linked together. Include all senior design requirements including design reports and Expo materials (i.e. Abstract, Poster, and Presentation). All major milestones should be included in the timeline.



Completed Task	Timeline (Start Date - End Date)
Team Contract:	September 1–6, 2024
Project Proposal:	September 7–October 11, 2024

<b><i>Fall Design Poster:</i></b>	<i>October 1–November 15, 2024</i>
<b><i>Preliminary Design Report:</i></b>	<i>November 16–December 9, 2024</i>

## Appendix 2: Team Contract (i.e. Team Organization)

### Step 1: Get to Know One Another. Gather Basic Information.

**Task:** This initial time together is important to form a strong team dynamic and get to know each other more as people outside of class time. Consider ways to develop positive working relationships with others, while remaining open and personal. Learn each other's strengths and discuss good/bad team experiences. This is also a good opportunity to start to better understand each other's communication and working styles.

<b><i>Team Member Name</i></b>	<b><i>Strengths each member bring to the group</i></b>	<b><i>Other Info</i></b>	<b><i>Contact Info</i></b>
Ahmad Scruggs	-Teamwork and Collaboration - Problem Solving - Time management -Communication	- Java, Python, C - Comfortable in git	<a href="mailto:scruggsaz@vcu.edu">scruggsaz@vcu.edu</a> (804)572-1658
Isaac Lyu	-Good communication - Timely Schedules -Brain storming ideas -Technical skills	- Java, python, C, - Comfortable in git	<a href="mailto:lyuid@vcu.edu">lyuid@vcu.edu</a> 7034037960
Matthew Baker	- good communicator - punctual	- online git and command online certification - java, javascript, C, and python	<a href="mailto:bakermj2@vcu.edu">bakermj2@vcu.edu</a> 423-493-3931

		experience	
Gamal Almareh	<ul style="list-style-type: none"> <li>- Time Management</li> <li>- Problem Solving</li> <li>- Active Listening</li> <li>- Attention To Detail</li> </ul>	<ul style="list-style-type: none"> <li>-Java, C, Python</li> <li>- Comfortable in git</li> </ul>	<a href="mailto:almarehgt@vcu.edu">almarehgt@vcu.edu</a> (917)564-4532

<i>Other Stakeholders</i>	<i>Notes</i>	<i>Contact Info</i>
Advisor: Tom Arodz		tarodz@vcu.edu
Sponsor: Mahesh Nair,  Vinay Soni		mahesh.bahulleyannair@capitalone.com  vinay.soni@capitalone.com

## Step 2: Team Culture. Clarify the Group's Purpose and Culture Goals.

**Task:** Discuss how each team member wants to be treated to encourage them to make valuable contributions to the group and how each team member would like to feel recognized for their efforts. Discuss how the team will foster an environment where each team member feels they are accountable for their actions and the way they contribute to the project. These are your Culture Goals (left column). How do the students demonstrate these culture goals? These are your Actions (middle column). Finally, how do students deviate from the team's culture goals? What are ways that other team members can notice when that culture goal is no longer being honored in team dynamics? These are your Warning Signs (right column).

**Resources:** More information and an example Team Culture can be found in the Biodesign Student Guide "Intentional Teamwork" page (webpage | PDF)

<i>Culture Goals</i>	<i>Actions</i>	<i>Warning Signs</i>
<i>Making sure work is split up fairly.</i>	<ul style="list-style-type: none"> <li>- Everyone has tasks that are agreed upon and be completed in timely manner</li> <li>- Decide who does what during meetings</li> </ul>	<ul style="list-style-type: none"> <li>- Team member misses a deadline for a task that was their responsibility and doesn't communicate it</li> </ul>
<i>Consistently meeting at least once a week</i>	<ul style="list-style-type: none"> <li>- Setting up weekly meetup events on Discord</li> </ul>	<ul style="list-style-type: none"> <li>- Team members consistently miss meetings and don't make</li> </ul>



	<i>- meet at 5 pm on thursdays during this semester in order to go over what we need to do</i>	<i>it up in some way.</i>
<i>Create a quality project</i>	<i>- Every team member is working to prioritize team goals - Making sure that each task is done with effort</i>	<i>- Team member does bare minimum rather than trying to make something worthwhile</i>

### Step 3: Time Commitments, Meeting Structure, and Communication

**Task:** Discuss the anticipated time commitments for the group project. Consider the following questions (don't answer these questions in the box below):

- What are reasonable time commitments for everyone to invest in this project?
- What other activities and commitments do group members have in their lives?
- How will we communicate with each other?
- When will we meet as a team? Where will we meet? How Often?
- Who will run the meetings? Will there be an assigned team leader or scribe? Does that position rotate or will the same person take on that role for the duration of the project?

**Required:** How often you will meet with your faculty advisor advisor, where you will meet, and how the meetings will be conducted. Who arranges these meetings? See examples below.

<i>Meeting Participants</i>	<i>Frequency Dates and Times / Locations</i>	<i>Meeting Goals Responsible Party</i>
<i>Students Only</i>	<i>As Needed, On Discord Voice Channel</i>	<i>- Update group on day-to-day challenges and accomplishments - assign new tasks - Check in with team - Isaac &amp; Matt as Scribe</i>
<i>Students Only</i>	<i>Every thursday at 5pm on</i>	<i>- Actively work on tasks that</i>

	<i>discord/at library depending on availability</i>	<i>require all team members - Critique each other's work - Discusses future planning</i>
<i>Students + Faculty advisor</i>	<i>thursdays at 4:45</i>	<i>Update faculty advisor and get answers to our questions Isaac or Matt will scribe</i>
<i>Project Sponsor</i>	<i>Thursdays at 5 for now but subject to change</i>	<i>Update project sponsor and make sure we are on the right track Isaac or Matt will scribe</i>

#### Step 4: Determine Individual Roles and Responsibilities

**Task:** As part of the Capstone Team experience, each member will take on a leadership role, in addition to contributing to the overall weekly action items for the project. Some common leadership roles for Capstone projects are listed below. Other roles may be assigned with approval of your faculty advisor as deemed fit for the project. For the entirety of the project, you should communicate progress to your advisor specifically with regard to your role.

- **Before meeting with your team**, take some time to ask yourself: what is my “natural” role in this group (strengths)? How can I use this experience to help me grow and develop more?
- **As a group**, discuss the various tasks needed for the project and role preferences. Then assign roles in the table on the next page. Try to create a team dynamic that is fair and equitable, while promoting the strengths of each member.

#### Communication Leaders

**Suggested:** Assign a team member to be the primary contact for the client/sponsor. This person will schedule meetings, send updates, and ensure deliverables are met.

**Suggested:** Assign a team member to be the primary contact for faculty advisor. This person will schedule meetings, send updates, and ensure deliverables are met.

#### Common Leadership Roles for Capstone

1. **Project Manager:** Manages all tasks; develops overall schedule for project; writes agendas and runs meetings; reviews and monitors individual action items; creates an environment where team members are respected, take risks and feel safe expressing their ideas.

**Required:** On Edusourced, under the Team tab, make sure that this student is assigned the Project Manager role. This is required so that Capstone program staff can easily identify a single contact person, especially for items like Purchasing and Receiving project supplies.

2. **Logistics Manager:** coordinates all internal and external interactions; lead in establishing contact within and outside of organization, following up on communication of commitments, obtaining information for the team; documents meeting minutes; manages facility and resource usage.

3. **Financial Manager:** researches/benchmarks technical purchases and acquisitions; conducts pricing analysis and budget justifications on proposed purchases; carries out team purchase requests; monitors team budget.

4. **Systems Engineer:** analyzes Client initial design specification and leads establishment of product specifications; monitors, coordinates and manages integration of sub-systems in the prototype; develops and recommends system architecture and manages product interfaces.

5. **Test Engineer:** oversees experimental design, test plan, procedures and data analysis; acquires data acquisition equipment and any necessary software; establishes test protocols and schedules; oversees statistical analysis of results; leads presentation of experimental finding and resulting recommendations.

6. **Manufacturing Engineer:** coordinates all fabrication required to meet final prototype requirements; oversees that all engineering drawings meet the requirements of machine shop or vendor; reviews designs to ensure design for manufacturing; determines realistic timing for fabrication and quality; develops schedule for all manufacturing.

<i><b>Team Member</b></i>	<i><b>Role(s)</b></i>	<i><b>Responsibilities</b></i>
<i>Matthew Baker</i>	<i>Scribe Systems Engineer</i>	<i>- Keep record of meetings we have, with important details and posts it for easy access by group - will work with sponsor to decide what is needed for project and acquire needed materials</i>
<i>Gamal Almareh</i>	<i>Manufacturing Engineer</i>	<i>- Make sure prototype meets requirements - Oversee requirements of the project and review designs - Communicate with Group on requirements and aspects of the project</i>
<i>Isaac Lyu</i>	<i>Scribe Logistics Manager</i>	<i>- Record key notes during meetings</i>

	<i>Project Manager</i>	<ul style="list-style-type: none"> <li>- Ensure that meetings are held consistently</li> <li>- Managing time efficiently</li> <li>- Communicating with internal and external relations</li> </ul>
<i>Ahmad Scruggs</i>	<i>Test Engineer</i>	<ul style="list-style-type: none"> <li>- Creates plans for testing</li> <li>- Writing and Executes test cases</li> <li>- Identifies and reports bugs in development</li> <li>- Documents test results</li> </ul>

**Step 5: Agree to the above team contract**

**Team Member:** *Isaac Lyu*

**Signature:** *Isaac Lyu*

**Team Member:** *Gamal Almareh*

**Signature:** *Gamal Almareh*

**Team Member:** *Ahmad Scruggs*

**Signature:** *Ahmad Scruggs*

**Team Member:** *Matthew Baker*

**Signature:** *Matthew Baker*

### **Appendix 3: [Insert Appendix Title]**

Note that additional appendices may be added as needed. Appendices are used for supplementary material considered or used in the design process but not necessary for understanding the fundamental design or results. Lengthy mathematical derivations, ancillary results (e.g. data sets, plots), and detailed mechanical drawings are examples of items that might be placed in an appendix. Multiple appendices may be used to delineate topics and can be labeled using letters or numbers. Each appendix should start on a new page. Reference each appendix and the information it contains in the main text of the report where appropriate.

**Note:** Delete this page if no additional appendices are included.

## References

Provide a numbered list of all references in order of appearance using APA citation format. The reference page should begin on a new page as shown here.

- [1] VCU Writing Center. (2021, September 8). *APA Citation: A guide to formatting in APA style*. Retrieved September 2, 2024. <https://writing.vcu.edu/student-resources/apa-citations/>
- [2] Teach Engineering. *Engineering Design Process*. TeachEngineering.org. Retrieved September 2, 2024. <https://www.teachengineering.org/populartopics/designprocess>
- [3] IBM Watson's AI in IT Service Management:  
**Source:** IBM. (n.d.). *IBM Watson for IT operations*. IBM.  
Retrieved October 10, 2024. <https://www.ibm.com/cloud/watson-aiops>
- [4] ServiceNow Virtual Agent:  
**Source:** ServiceNow. (n.d.). *ServiceNow virtual agent*. ServiceNow.  
Retrieved October 10, 2024. <https://www.servicenow.com/products/virtual-agent.html>
- [5] IPsoft's Amelia for ITSM:  
**Source:** IPsoft. (2020). *Amelia for IT Service Desk*. IPsoft.  
Retrieved October 10, 2024. <https://amelia.com/it-service-desk/>
- [6] AI in IT Service Management:  
**Source:** Gartner. (2019). *Hype cycle for IT service management*. Gartner.  
Retrieved October 10, 2024. <https://www.gartner.com/en/documents/3880910>