Presenter, company name and addressTitle and subtitleVersion number and dateDecorative sidebar

Table of Contents

[Project Description 2](#_heading=h.gjdgxs)

[Overview (ABET-2): 2](#_heading=h.30j0zll)

[GLOBAL TRENDS (EM@FSE-e): 2](#_heading=)

[MARKET ANALYSIS (EM@FSE-k): 2](#_heading=)

[SECURITY CONSIDERATIONS (SER-2): 3](#_heading=)

[Key Requirements (SER-2): 3](#_heading=h.1fob9te)

[Deliverables (SER-1): 3](#_heading=h.3znysh7)

[Acronyms and abbreviations (ABET-3): 3](#_heading=h.2et92p0)

[Design and Architecture 4](#_heading=h.tyjcwt)

[Design Description (ABET-1, ABET-2): 4](#_heading=h.tz000q9ed3bg)

[Alternate Design Possibilities (EM@FSE-b) 5](#_heading=h.xzbdgbw261eh)

[Implementation Strategy 5](#_heading=h.3dy6vkm)

[High-level Work Breakdown Structure (SER-1): 5](#_heading=h.1t3h5sf)

[Schedule / Timeline (SER-1): 7](#_heading=h.4d34og8)

[Required Hardware (SER-1, EM@FSE-o): 9](#_heading=h.2s8eyo1)

[THird party content (SER-1, EM@FSE-o): 10](#_heading=h.17dp8vu)

[Quality (SER-2): 10](#_heading=h.3rdcrjn)

[References/Sources of Information (EM@FSE-q): 11](#_heading=)

[Scalability (EM@FSE-J): 12](#_heading=)

[Other Special considerations (ABET-2): 13](#_heading=h.lnxbz9)

[process 13](#_heading=h.35nkun2)

[Process Description and justification (SER-1) 13](#_heading=h.1ksv4uv)

[Tools (SEr-1, EM@FSE-O): 13](#_heading=h.44sinio)

[Roles and Responsibilities (SER-1): 14](#_heading=h.2jxsxqh)

[Location of Project Artifacts (SER-1): 14](#_heading=h.z337ya)

[Sponsor communications (ABEt-3): 14](#_heading=h.3j2qqm3)

[identified Potential risks (SER-2): 15](#_heading=h.4i7ojhp)

[mitigation strategies (SER-2): 17](#_heading=h.2xcytpi)

# **Project Description**

## Overview (ABET-2):

The application the client requested will provide a means for members of various volunteer organizations to report vital information to emergency responders during natural disasters. The project aims to address the challenge of conveying this information during crises when cellular data and internet might be unavailable. In the past, this information was relayed on paper, which extended the time needed for transportation and analysis of the data. The final application will feature a questionnaire for users to complete. This data will then be saved locally on the user's device and must be available for extraction via USB to an external device, ensuring it can be hand-delivered to emergency responders.

The expected users of this application are members of the volunteer organizations MYN and CERT. However, the app will also be accessible to the general public. It will be available on both Android and iOS platforms. The primary motivation for creating this application is to overcome the limitations of the current app, which is only available on Android and requires an internet connection to transmit essential data.

## GLOBAL TRENDS (EM@FSE-e):

This project fits well with the trend for data and data research, where large volumes of data are necessary to develop an image of the state of an object or group of objects. Data on the conditions of individual residences would in this case paint the picture for the severity and economic impact of a natural disaster.

Health is also a theme of this project, where the humanitarian efforts also are the primary means for getting aid to those in need. The volunteer in the field is the primary point of contact for the patients, and is the one performing the assessment of their wellbeing. The triage decisions are made by those who collect the data from all the volunteers. Without good and timely data, the triage decisions are not as effective.

## MARKET ANALYSIS (EM@FSE-k):

There are a few alternatives to what the client is asking for. For instance the client could simply use paper copies of the FEMA forms he wants to create digitally in the app (<https://www.fema.gov/about/openfema/data-sets/national-household-survey>). This solution would be quite slow as all the results would have to be collected manually on paper and put into a computer manually leading to multiple points of user error ruining the data collected and slowing down the disaster relief. Another Solution would be what the organization currently uses, a modified KoboCollect app. This however has flaws as well from being a bit clunky to use for the user, requiring internet connection and only being available on Android. Overall there isn’t a one size fits all solution on the market and the best solution for our client would be the development of a custom application.

## SECURITY CONSIDERATIONS (SER-2):

Device security will be important for protecting the device owner and the information of those being gathered on the device. The application must not allow access to secure portions of the device. The device owner’s personal information, banking information, or any other information normally used on the phone or tablet would be at risk if the application were not secure.

The information collected by the application regarding the conditions at a natural disaster site (such as names, addresses, health conditions, etc.) could be considered PPI or health PPI, and would need to be securely stored and transmitted.

## 

## Key Requirements (SER-2):

* The application shall be cross platform (iOS and Android compatible).
* The application shall save information from the host device to a USB stick
* The information collected by the device shall be from the authorized FEMA documents.
* The application shall automate getting date/time and GPS location information.
* The application must include data validation checks to ensure the accuracy and completeness of collected information.
* ~~The application must seamlessly integrate with the county GIS mapping system.~~
* The application shall require mandatory questions be answered prior to saving a report
* The application will not automatically fill in default answers for human input questions.
* The application must allow editing of the answers to previous questions prior to saving.
* The application will save information in the ‘CSV’ format.

## Deliverables (SER-1):

* Individual training on required technologies and frameworks
* Provide sponsor with user interface prototype
* Acquire licenses necessary to begin implementation in Android and iOS environments
* Develop Backend Infrastructure
* Complete Android version of application, receive feedback from sponsor
  + Offline Data Storage and Synchronization
  + Able to fetch accurate GPS and Geo-location
* Complete iOS version of application, receive feedback from sponsor
* Make changes based on feedback (as necessary)
* Complete testing and debugging
* Support and maintenance plan

## 

## Acronyms and abbreviations (ABET-3):

* FEMA - Federal Emergency Management Agency
* GPS - Global Positioning System
* GIS - Geographic Information System
* USB - Universal Serial Bus
* USB-C - Universal Serial Bus Type C
* JSON - JavaScript Object Notation
* OS - Operating System
* MYN - Acronym for Map Your Neighborhood
* CERT - Stands for Civilian Emergency Response Team

# 

# **Design and Architecture**

## Design Description (ABET-1, ABET-2):

The Application (App):

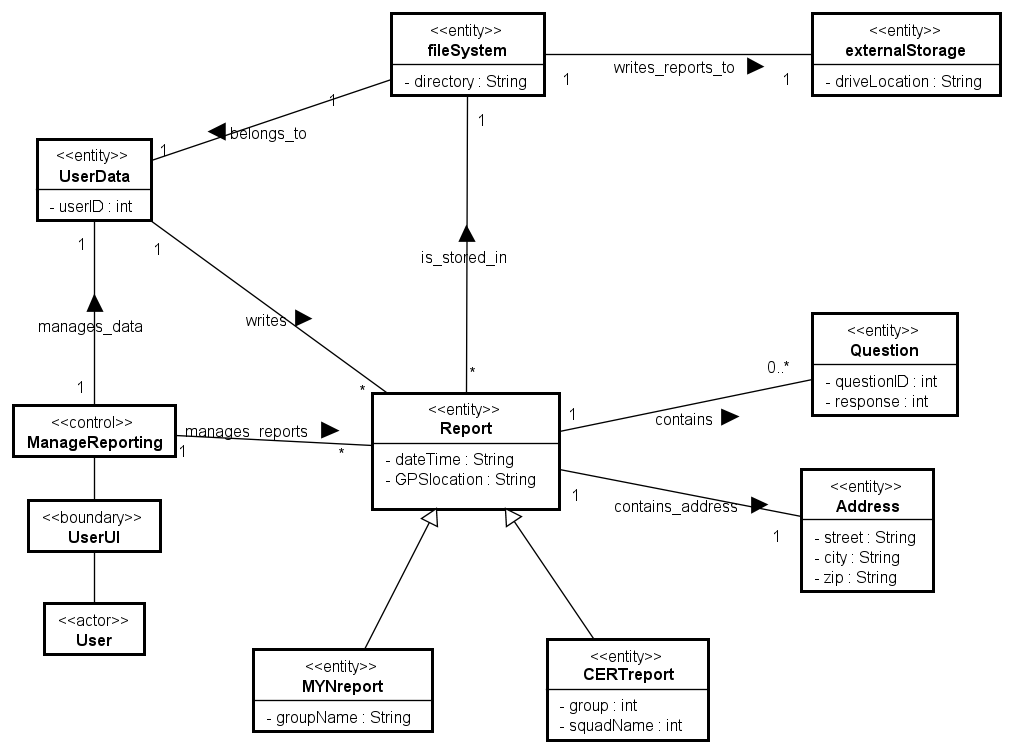
The App itself will be responsible for accepting user inputs, converting those inputs to a JSON message, and storing that message in a file. The App should be easy to use and understand. GPS positional data dn time stamps should be automatically collected by the App when the user creates a new report. The App will also be responsible for moving the data on the host device to the USB storage when prompted.

Device storage/file (database):

The file system is responsible for housing the JSON data until moved to an external storage device. The data at rest and on the move needs to be secure and unchanged.

External USB Drive:

The USB is only responsible for securely storing the data offloaded from the device.



## Alternate Design Possibilities (EM@FSE-b)

The current strategy being used for this process involves multiple separate applications and is not straight forward. The team decided consolidating all the tasks necessary to achieve the goal into one easy to use application was a better solution. Since the scope of this project is the device application itself and not the distributed system as a whole, the single application performing all necessary functions makes the most sense.

# 

# 

# **Implementation Strategy**

## High-level Work Breakdown Structure (SER-1):

* Individual training on required technologies and frameworks

Tasks will include each team member completing a Udemy course on React Native and doing individual research on the required operating systems. This will require some background knowledge of Javascript and should take no more than two weeks to complete.

Some time will be spent completing tasks to aid in learning how to work with both iOS and Android development environments. A potential risk for this deliverable is the uncertainty of how long this learning could take at the time of writing this document, and there may be specific licenses required to begin working in the specified environments.

* Provide sponsor with user interface prototype

Team members will be responsible for completing tasks that consist of different sections of the application. This will require some basic knowledge of UI design and will be completed using the Pencil application.

* Develop Backend Infrastructure

Tasks will consist of developing varying sections of the codebase for the app relating to background structure. These tasks will include writing code to handle UI interaction logic, compile data into the requested format, write to a local file system, and extract data to an external storage device. Risks associated with these tasks include handling compatibility with both Android and iOS and completing work within the projected timeline.

* Complete Android version of application, receive feedback from sponsor

Tasks will include developing frontend source code based on previous feedback on the original UI prototype, and all other tasks associated with finalizing a working application in the Android environment. Previous knowledge will include the research done in the first deliverable. Risks associated with these tasks include possible changes to the prototype requested by the sponsor that will extend the timeline for this deliverable, and acquiring proper licensing to deploy the app.

* Complete iOS version of application, receive feedback from sponsor

Tasks will include making any compatibility changes to source code necessary to deploy the application in the iOS environment. Previous knowledge will include the research done in the first deliverable. Risks involve requirement changes that require revisiting the previous deliverable, which would cause a delay in the overall projected timeline. This deliverable is also affected by needing specific licenses for deploying the application.

* Make changes based on feedback (as necessary)

Tasks for this deliverable include fulfilling any requested changes made by the sponsor, resulting in a working product with functionality that the sponsor is satisfied with. A large risk associated with this deliverable is that it is iterative, and how much time can be given to this deliverable is largely dependent on previous deliverables being completed on time.

* Complete testing and debugging

Tasks will include developing and implementing field and stress tests on the application. A major risk for this deliverable is having necessary hardware available to conduct field testing, and knowledge of how much stress testing is required for the use cases of the application.

* Support and maintenance plan

Tasks will include meeting with the sponsor to discuss support for the application going forward after the project timeline is complete. Tasks for this deliverable will largely be dependent on the sponsor’s decisions on how to handle maintaining the application beyond the scope of the development project.

## 

## Schedule / Timeline (SER-1):

Our project schedule spans two academic semesters, Fall and Spring, to ensure comprehensive development and testing. The timeline is constructed with input from the sponsor and considers potential project activity during the winter break.

**Sep-Nov: Training/Design/Predev (including UI Design)**

**Week 0: Pre-planning, Requirements gathering**

* Members will complete Udemy React Native Training
* Set project vision, project requirements

**Week 1 - 2: Oct. 1st Project Kickoff**

**Deliverables:**

* Basic understanding of React by all members of the team
* Class Diagram
* Use-case Diagrams

**Week 3 - 4: Design & Prototyping**

**Deliverables:**

* Finalization of UI design
  + Sequence Diagrams
  + Activity Diagrams

**Nov-Feb: Development Stage**

**Week 5 - 6: Development - Alpha Prototype**

**Deliverables:**

* Implement core functionalities.
* Prepare for initial testing.

**Week 7 - 8: Testing - Phase 1**

**Deliverables:**

* App is able to guide user through forms
* App will generate a JSON file based on user results and save it to the phone.

**Week 9 - 10: Development - Phase 1**

**Deliverables:**

* App will be able to detect thumb drive inserted into phone
* App will be able to automatically export JSON file to USB stick

**Week 11 - 12: Testing - Phase 1**

**Deliverables:**

* Android prototype with working UI and basic features
* iOS prototype with working UI and basic features.

**Week 13 - 14: Development - Phase 2**

**Deliverables:**

* Ability to link pictures to the generated output message
* GPS data automatically added to JSON messages
* Timestamps automatically added to JSON message

**Week 15 - 16: Testing - Phase 2**

**Deliverables:**

* App will be able to store photos taken through the app and link to a specific JSON entry.
* App will export out photos to the USB drive automatically

**Week 17 - 18: Final Development**

**Deliverables:**

* Functionally complete App on both IOS/Android.

**Week 19 - 20: Additional Requirements addons**

**Deliverables:**

* 5 icon png images in the required sizes for Android
* 7 icon png images in the required sizes for iOS

**Mar: Functionality Testing**

**Week 21 - 22: Quality Control**

**Deliverables:**

* An Android app that meets all of google’s requirements in order to be uploaded to the play store
* An iOS app that meets all of Apple’s requirements in order to be uploaded to their store.

**Week 23 - 24:**

**Deliverables**:

* Tested product
* Ability to save to an external device
* Ensure product meets requirements for IOS and Google stores

**Apr: Finalization**

**Week 25 - 26:**

**Deliverables:**

* Working Product ready to be deployed.

**Week 27 - 28: Buffer Period**

## Required Hardware (SER-1, EM@FSE-o):

1. **USB-C Data Stick:** We will need a USB-C data stick to test and demonstrate the functionality of transferring data from the mobile application to another external device.
2. **USB-C Cable (Phone/Tablet to USB-C):** USB-C cables will be required for connecting mobile devices (both Android and iOS) to the USB-C data stick for data transfer testing.
3. **Android Phone/Tablet:** Android devices will be necessary for development, testing, and demonstrating the application's functionality on the Android platform. Team members may use their own devices, and if necessary, additional devices can be acquired for testing purposes.
4. **iPhone/iPad:** iOS devices (iPhone/iPad) will be required for iOS app development, testing, and demonstration. Team members may use their own iOS devices, and if needed, additional devices can be obtained for testing purposes.
5. **Mac OS Device (for iOS Development):** A Mac OS device will be necessary for iOS app development using Xcode, which is the official development environment for iOS. This device will be used during the development phase.

## THird party content (SER-1, EM@FSE-o):

**Third-Party Content:**

1. **GPS Data Gathering Plugin:** To collect GPS data within the application, we may use a third-party plugin or open-source library, such as a Google Maps plugin. The licensing terms for this third-party content will be carefully reviewed to ensure compliance with usage rights and licensing agreements. Any obligations related to the use of this content will be followed, including proper attribution and adherence to licensing terms.
2. **GIS (Geographic Information System) Services:** Geographic Information System services may be utilized for mapping and spatial analysis within the application. Depending on the specific GIS services employed, we will ensure compliance with their licensing terms and usage policies. Proper attribution and adherence to licensing agreements will be maintained when incorporating GIS data and services into the application.

## 

## Quality (SER-2):

**Quality Goals:**

1. **Reliability/Performance:** The app should always work well, even in tough situations like disasters. It shouldn't suddenly crash or stop working.
2. **Usability:** The app should be easy to use and straightforward, so even new volunteers can easily install and use the application without much training. The design requirements also play into the usability, and must be adhered to.
3. **Data Integrity:** The info the app collects needs to be correct and safe. The data directly aids in getting help and supplies to those in need. Any error in the data from processing could extend the time required for aid to be received. It is important that the data remain valid from inception to delivery of the data.
4. **Cross-Platform Compatibility:** The app should work the same way on both Android and iOS devices. Each required functionality shall be provided in both versions.
5. **Security:** Private information regarding addresses and health info will be handled and stored by the application. This information must not be accessible to those who might use it for exploitation purposes.

**Quality Assurance Strategy:**

1. **Testing and Quality Assurance:** Rigorous testing will be performed, including testing the application across different environments and phases of development. The customer will have high visibility of the product to ensure the quality is of the desired caliber.
2. **User-Centered Design:** The sponsor will be included in user testing to ensure the usability meets the customer’s standards and expectations.
3. **Cross-Platform Compatibility:** The app should perform the same functions on Android and iOS. To ensure this, the same tests will be run against both versions, with the same input parameters to ensure the desired output from each version is the same.
4. **Code Quality and Readability:** Code will be well structured and commented for easier understanding. Code reviews will be performed to ensure readability and compliance to coding standards.

## References/Sources of Information (EM@FSE-q):

**React/React Native for Cross-Platform Development**:

React native is currently the planned backbone of our application. React was chosen for its cross platform compatibility. The members of our team are new to using React and will require quite a bit of research when starting out.

* <https://www.udemy.com/>
* <https://reactnative.dev/>
* <https://github.com/facebook/react-native>

**Offline Data Storage and Synchronization**:

As the disaster situation can lead to lack of internet connectivity, understanding offline data storage mechanisms and how to synchronize them when connectivity is restored is crucial.

* [React Native Offline First Apps: A Step-by-Step Tutorial](https://www.bacancytechnology.com/blog/react-native-offline-support#:~:text=You%20can%20use%20local%20databases,connection%20is%20available%20for%20synchronization.)
* [Creating an offline-first React Native app](https://blog.logrocket.com/creating-offline-first-react-native-app/)

**Fetching GPS/Geo-location Coordinates**:

Accurate geolocation is crucial in emergency situations for the pinpointing of hazards, teams, and affected areas. Understanding how GPS works, especially in disaster scenarios, can enhance app efficiency.

* [Geolocation - React Native Archive](https://archive.reactnative.dev/docs/geolocation#:~:text=Geolocation%20is%20enabled%20by%20default,'Capabilities'%20tab%20in%20Xcode.)
* [React Native geolocation: A complete tutorial](https://blog.logrocket.com/react-native-geolocation-complete-tutorial/)

## Scalability (EM@FSE-J):

Revenue streams:

* This project will be open source/non-profit, and supported by non-profit organizations. Given the app’s utility in disaster management, it might be possible to secure grants or funds from local, state, or federal agencies focused on emergency management.

Key Partners:

* Work closely with county and city officials to promote the use of the app, gather feedback and ensure it aligns with local requirements. Partnerships with MYN, CERT, and similar organizations.

Costs:

* License fee to publish app in android and ios app store.

Key Resources:

* A dedicated technical team for app development, and maintenance. And, a community manager/sponsor to engage with users, gather feedback, and ensure effective communication between all stakeholders.

This project focuses on the capabilities of a single device (phone/tablet) at a time. The data produced by the application goes to a database outside the scope of our work. If the project were to be scaled up, the database and collection methods would likely need to be included. The number of volunteers collecting data is ultimately limited by the amount of data the database is able to receive. Streamlining the data collection (over wireless ethernet or cellular connection) would be a significant step in solving the larger issue of time required to collect and analyze the data. Working with humanitarian organizations for funding would go a long way in farthing that development.

The possible constraints could be the platform compatibility. The project began with a focus on Android, and if the iOS platform isn’t quickly adapted, it might alienate half the potential user base. Our team will invest in cross-platform development tools, and frameworks, such as React Native, or Flutter. Data integrity could also be an issue, because the life saving nature of the application, implementing robust data verification and reconciliation mechanisms is essential. Additionally, employ data integrity checks when syncing offline data.

## 

## Other Special considerations (ABET-2):

The format of the data being sent to the database was not specified by the customer and is not known. The team for now is assuming JSON format messages will be acceptable, or able to be converted to an acceptable format.

# 

# 

# **process**

## Process Description and justification (SER-1)

The process the team will be using is the Agile Scrum method. The Scrum method provides a workflow structure for efficient task completion and collaboration while allowing for flexibility in regards to the design and maturation of the product. Scrum projects focus on periods of quick development, then short periods of reflection and refocusing. These periods of refocusing will be good points to include the sponsor and ensure the product is in line with their requirements. The Scrum method also is viable for asynchronous work, only requiring members to meet for specific short meetings at the end of each sprint.

## 

## Tools (SEr-1, EM@FSE-O):

* **Version Control Systems**:
  + **Example**: [Git](https://git-scm.com/) ([GitHub](https://github.com/), GitLab, Bitbucket)
  + **Purpose:** Track and manage changes to codebase, facilitate collaboration among developers, and maintain different versions of the software.
* **Project Management and Collaboration Tools**:
  + **Example:** [Taiga](https://taiga.io/) (Scrum process), [Discord](https://discord.com/) (Team Communications), [Google Docs](https://www.google.com/docs/about/) (Documentation).
* **Design and Mockup Tools**:
  + **Example:** [LucidChart](https://www.lucidchart.com/pages/) (Diagrams), [Adobe XD](https://helpx.adobe.com/support/xd.html), [Figma](https://www.figma.com/)
* **Integrated Development Environments (IDEs)**
  + **Example**: [Android Studio](https://developer.android.com/studio), [Xcode](https://developer.apple.com/xcode/), [Visual Studio](https://visualstudio.microsoft.com/).
  + **Purpose**: They offer comprehensive facilities to software developers for software development, including editor, compiler, debugger, etc. everyone is free to choose their own IDEs for development.
* **Frameworks**
  + **Example:** [React Native](https://reactnative.dev/), [Flutter](https://flutter.dev/)
  + **Purpose**: Cross-Platform Development, write once and deploy across platforms.

## 

## Roles and Responsibilities (SER-1):

* **Scrum Master**: Guides the team in performing effective Scrum meetings and sprints.
* **Product Owner**: Ensures the Team’s focus is in line with the needs of the customer, and the product backlog reflects the requirements set forth by the team and customer.
* **Github Owner**: Responsible for organization of the Github Repository and responsible for the merges from the development branch to the main branch. All development will be performed from branches off of the development branch.
* **Sponsor Liaison:** Ensures the sponsor is an active member of the development and is the main point of contact for asking questions and relaying this information to the team.

## 

## Location of Project Artifacts (SER-1):

* **Github**: [(repository)](https://github.com/dchanley/SER401Team15)
* **Taiga**: [(Scrum process)](https://tree.taiga.io/project/clhughe9-emergency-response-data-collection-app/timeline)
* **Google Docs**: [(Documentation)](https://drive.google.com/drive/folders/1rYAXsf8pAh8CTEUG-iQBxNDobRqIyRR3?usp=sharing)

## 

## Sponsor communications (ABEt-3):

[David Hanley](mailto:dchanley@asu.edu) will be the primary POC with the sponsor. Based on current communications, the next available meeting times with the sponsor will be determined at the end of each sponsor meeting, or via email. This is subject to change based on the sponsor’s needs. Sponsor meetings will be held over Zoom. We will maintain a section of our discord where we will post questions asked to the sponsor and his answers as we get them to make sure the team has a full history of what he wants.

**Risk management**

## identified Potential risks (SER-2):

#### Group Dynamics:

* 1. **Cohesion**: Varying experiences and skills can affect team cohesion. Misunderstandings might arise if not everyone is on the same page. A majority of our team members lack extensive experience in Phone App development. Consequently, there will be a learning curve, potentially leading to mistakes or inefficiencies in the project.
     + **Impact**: Delays in decision-making, inconsistent work quality, and a potential for missed deadlines.
     + **Incidence Rate**: High
  2. **Communication**: With members in different time zones, there's a risk of miscommunication or delayed information sharing. Our team members are located in different time zones across the country. This can complicate scheduling synchronous meetings and may result in delayed responses.
     + **Impact**: Delays in decision-making.
     + **Incidence Rate**: High.
  3. **Availability**: Due to other course commitments, team members might not always be available, risking delays in task completion. Our team consists of students concurrently enrolled in other courses. Effective time management is crucial, but balancing a full-time project of this magnitude with homework from other subjects can be challenging.
     + **Impact**: Delays in decision-making, inconsistent work quality, and a potential for missed deadlines.
     + **Incidence Rate**: High.
  4. **Sponsor Communication**: Inconsistent or unclear communication with sponsors or stakeholders.
     + **Impact**: Misunderstood project requirements, objectives, or expected deliverables.
     + **Incidence Rate**: Low. Scheduled and structured communication with stakeholders can reduce this risk.

#### Infrastructure:

* 1. **OS Cross-compatibility issues**: Functionalities implemented in one OS may not be directly cross-compatible with another OS or may exhibit different behaviors. This necessitates specific development tailored to each OS, thereby doubling the work and time required for each added functionality. While it's not anticipated that this will be a frequent issue, given that the team is unfamiliar with the resources in use, it remains a possibility.
     + **Impact**: Developing for both iOS and Android can be challenging due to different platform requirements and possible behavior discrepancies.
     + **Incidence Rate**: High. Cross-platform development often encounters this issue.
  2. **Hardware Access and Compatibility**: Inconsistent access to or compatibility with necessary hardware, especially considering the need for both iOS and Android platforms.
     + **Impact**: Inefficient testing and implementation processes, potentially leaving out a portion of the intended users.
     + **Incidence Rate**: Low. Our team will try to secure both platform devices for development and testing.
  3. **Dependency on internet infrastructure**
     + **Impact**: Given the nature of the app's usage during emergencies, reliance on stable internet connections for critical operations could be problematic.
     + **Incidence Rate**: Low. The context suggests solutions for offline functionality, but it's a consideration for any updates or new functionalities.

#### Design Limitations:

* 1. The current design has a number of limitations which could prevent efficient data collection and sharing.
     + **Impact**: If not addressed, these limitations might lead to inefficient data collection or miscommunication in emergencies.
     + **Incidence Rate**: High, given the current system's described limitations.

#### Implementation Blockers:

* 1. Encountering unforeseen technical challenges during the development phase.
     + **Impact**: Delays in the project timeline, increased costs, and potential sub-optimal workarounds.
     + **Incidence Rate**: Moderate. Software development projects often encounter unexpected challenges.

#### Incomplete Requirements:

* 1. Not all requirements or needs might be captured in the initial phases.
     + **Impact**: Future changes or revisions to the app, leading to potential inefficiencies and rework.
     + **Incidence Rate**: Moderate. Initial requirements gathering may miss some nuanced needs of the end-users.
  2. **Shifting goalposts**: As the product evolves and becomes more defined, there may be additional direct requirements from the customer. Such changes are anticipated, as they will serve to refine the final product once its capabilities are better understood by the customer. These late-stage requirements shouldn't alter the application's core functionalities and, therefore, are not expected to significantly impact the project's completion timeline.
     + **Impact**: Without a clear and consistent vision, the project can suffer from scope creep, causing delays and possibly leading to a product that doesn't fully meet needs.
     + **Incidence Rate**: Low. Given the detailed context provided, it's less likely, but it's always possible requirements will evolve.

## mitigation strategies (SER-2):

#### Group Dynamics:

* 1. **Cohesion**
     + **Mitigation**: Regular team-building exercises can be conducted to ensure team cohesion. Weekly knowledge-sharing sessions can be organized where team members share their experiences and learnings regarding phone app development.
     + **Cost**: Extra hours for team-building sessions; however, this can lead to better teamwork and fewer mistakes in the long run.
  2. **Communication**:
     + **Mitigation**: Set fixed times for meetings that accommodate all time zones, even if it means some meetings occur outside of standard work hours. Use a shared calendar tool to track team members' availability.
     + **Cost**: Some members might have to attend meetings at unconventional times, but this ensures everyone is informed and aligned.
  3. **Availability**:
     + **Mitigation**: Maintain a shared schedule or task management system that allows team members to allocate hours they can work on the project. This system can also track deliverables and deadlines to manage workload better.
     + **Cost**: Time spent on schedule management; however, it helps in ensuring tasks are completed on time.
  4. **Sponsor Communication**:
     + **Mitigation**: Set up bi-weekly or monthly check-ins with the sponsors. Document all meetings meticulously and share minutes with both team members and sponsors.
     + **Cost**: Time spent on organizing, conducting, and documenting meetings.

#### Infrastructure:

* 1. **OS Cross-compatibility issues**:
     + **Mitigation**: Use cross-platform development tools like React Native. These tools allow for simultaneous development for both iOS and Android. A Udemy course on the framework has been provided to the team to give everyone a good understanding of the fundamentals. We have also already expressed if any questions about how to do something comes up to have them posted in the discord so we can tackle these as a team.
     + **Cost**: Initial learning curve for these tools, but saves time in the long run.
  2. **Hardware Access and Compatibility**:
     + **Mitigation**: Secure devices for both iOS and Android for development and testing purposes. If purchasing is not feasible, consider device rental or emulators.
     + **Cost**: Budget allocation for devices or rental fees.
  3. **Dependency on internet infrastructure**
     + **Mitigation**: Design the app to function offline as much as possible, and sync when an internet connection is available.
     + **Cost**: Additional development time to implement offline functionalities.

#### Design Limitations:

* 1. The current design has a number of limitations which could prevent efficient data collection and sharing.
     + **Mitigation**: Seek feedback from potential users early in the design phase. Prototype testing can help in identifying and addressing design inefficiencies.
     + **Cost**: Time and possibly money spent on user testing and feedback sessions.

#### Implementation Blockers:

* 1. Encountering unforeseen technical challenges during the development phase.
     + **Mitigation**: Use an iterative development approach, like Agile, to address issues as they arise and adapt to changes. Regularly consult with experts or senior developers to guide through technical challenges.
     + **Cost**: Regular iterations might seem time-consuming, but they often prevent larger issues down the road.

#### Incomplete Requirements:

1. Not all requirements or needs might be captured in the initial phases.
   * **Mitigation**: Implement continuous feedback loops with stakeholders. Develop a MVP (Minimum Viable Product) first, to refine and add features based on actual user feedback.
   * **Cost**: Additional iterations based on feedback can impact the timeline, but result in a more refined product.
2. **Shifting goalposts**:
   * **Mitigation**: Set a clear change request process where any alterations to the project scope are documented, reviewed, and approved.
   * **Cost**: Time spent in evaluating and implementing changes. However, maintaining a controlled process prevents unchecked scope creep and manages stakeholder expectations.