

Architecture Design Document

Part II

**Contents**

[1.](#_gjdgxs) OVERVIEW 2

[2.](#_30j0zll) SOFTWARE ARCHITECTURE 2

[3.](#_1fob9te) USER EXPERIENCE / USER INTERFACE ARCHITECTURE 7

[4.](#_3znysh7) SECURITY ARCHITECTURE 11

[5.](#_2et92p0) INFRASTRUCTURE/DEPLOYMENT ARCHITECTURE 11

[6.](#_tyjcwt) TECHNOLOGY STACK 14

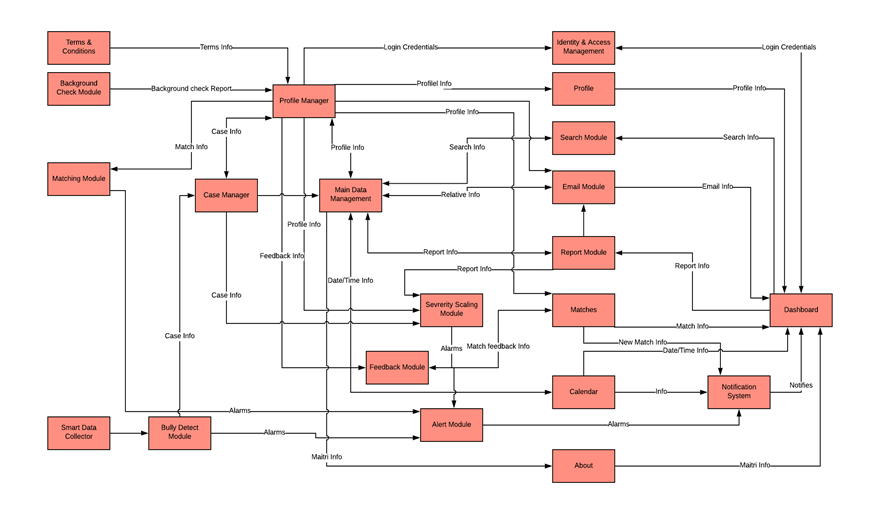
[7.](#_1t3h5sf) Glossary 15

[8.](#_2s8eyo1) References 15

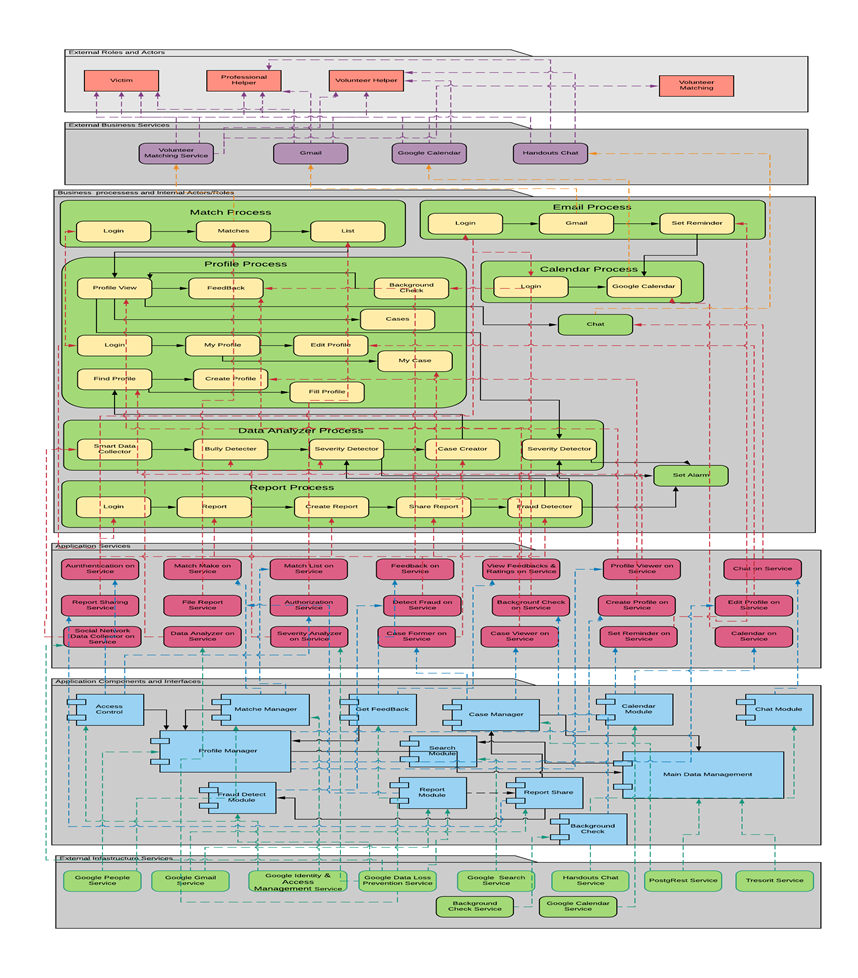
# OVERVIEW

This is the second part of the Architecture Document. This document consists description of Software Architecture, User Experience/ User Interface, Security Architecture, Infrastructure /Deployment Architecture, and Technology Stack. The following section gives the description on the respective topic with diagrams.

# SOFTWARE ARCHITECTURE

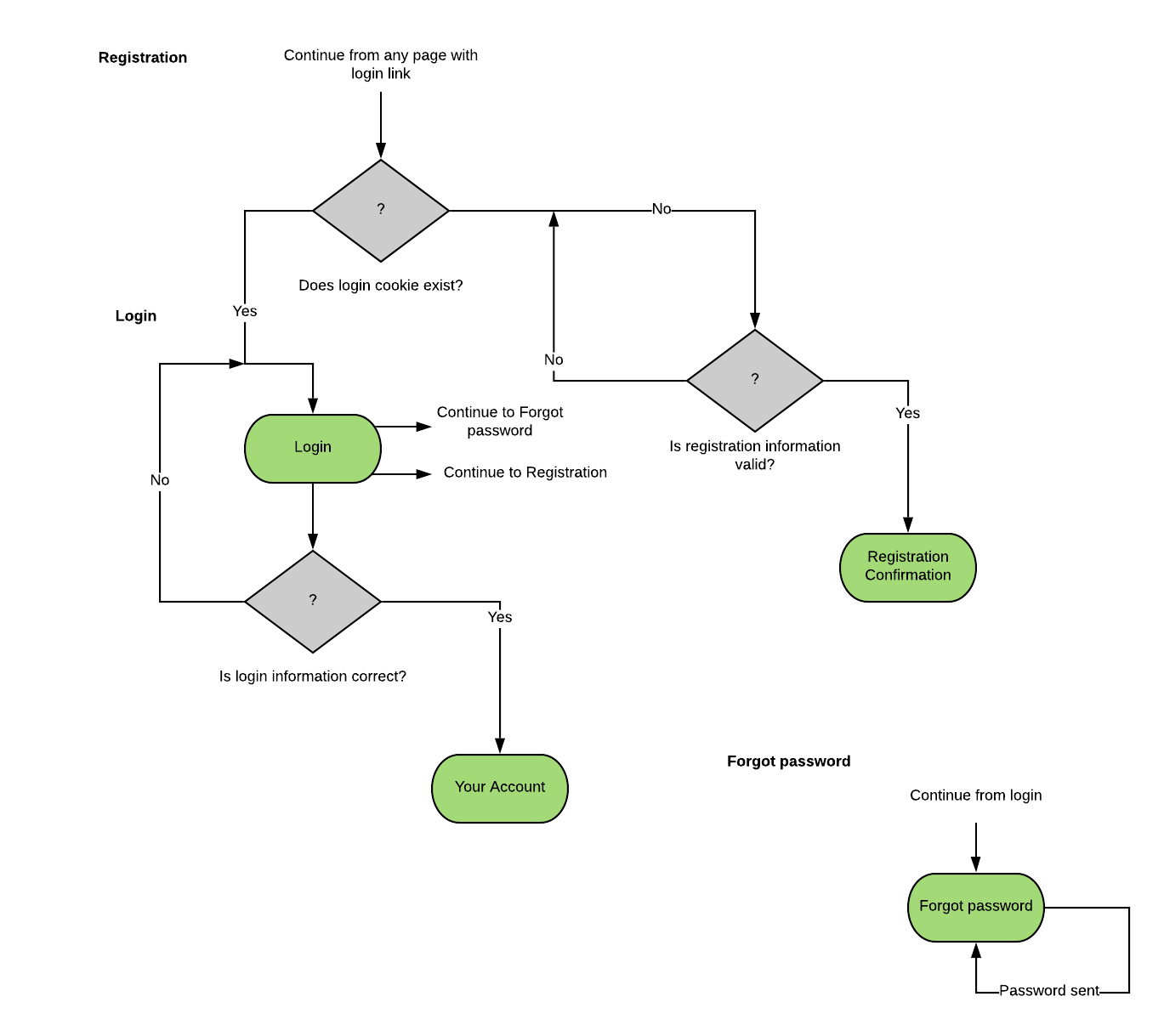


|  |  |  |
| --- | --- | --- |
| Component | Description | Key Non-Functional Qualities that are relevant |
| Terms & Conditions | This component makes sure that Terms and Condition are being viewed and agreed by the users before making an account or else just filing report. |  |
| Background Check Module | This component makes sure that volunteers/helpers and professional/legal helpers’ background has gone through legal check and makes sure its clear or else won’t let them to participate in any kind of helper role. | Security, Reliability |
| Matching Module | This component extracts the info from volunteer matching web application by integrating with them and also from the profile component of all the existing volunteers and helps and matches them with the victims so that their personality and level of severity matches and are matched accordingly. | Reliability, Availability |
| Profile Manager | This component is one of the most important components. It takes care of the profile data and manages their info by interacting with the main data management component. This component keeps all the data related with the personal in order so that it is easy to access. Some important data retrievals are cases, reports, background checks, credentials, authorizations and many more. | Performance, Usability |
| Case Manager | This component takes care of all the cases, recording all the bully scenarios from reports and any detected bully from social website through smart data collector component and feeds main data management component with its data. | Performance, Usability |
| Main Data Management | This component is responsible for all data to be organized and kept in order so that the components can extract from it and use it respect of their suit. This component uses POSTGRESQL. It directly stores in Tresorit cloud. | Performance, Usability, Availability |
| Smart Data Collector and Bully Detect Module | This component extracts data from social media using API from google. And then sends it to bully detect module component to detect any signs of bully using another API specialized for detecting data like words, patterns, etc. | Reliability, Availability |
| Feedback Module | This component gets in feedback and ratings from the users for their match partners and records I their profile for better future match and also help the helpers to where to focus in order to help the victim. Also, helps the victim to know ahead of time if the helper is good or not. | Reliability |
| Alert Module | This component is responsible to notify the user if there is a new match or the admin users of there is a new bully detection. |  |
| Severity Scaling Module | This component scales the severity of the patients over healing process, level of bully detected and reported and gives it a number. If the number is to high it sends a short info to alarm component to notify the admin user. | Reliability |
| Identity & Access Management | This component checks the user credentials and allows access to the system. | User access |
| Profile | This component allows user to see their own file and edit them. This component directly interacts with the profile manager component. |  |
| Search Module | This component helps the users to search for any data within the system. | Availability |
| Email Module | This component is directly integrated with Gmail through Gmail API. This component provides all the basic email features to the users so that they communicate with each other. | Usability, APerformance |
| Report Module | This component is the only component that can be accessed without even logging in. This event is mainly for either a witness or victim to file any report related to bully. The report module component sends the report for severity scale check and also stores it to main data. And if the victim related to the report is no found through the profile manager component then it automatically makes a new account for the new victim. | Usability, Availability, Performance |
| Matches | This component is responsible for making matches between victims and helpers. By doing this it makes sure that the matches are made on mostly severity scale. | Usability, Availability, Performance |
| Calendar | This component directly integrates with google calendar using cloud calendar API. This component helps the users to set reminders and keep track of meeting as it automatically prompts the notification system component to remind the users. | Usability, Performance |
| About | This component shares the information about the Maitri and ABS system general information if the user wants to know. |  |
| Notification System | This component notifies the users with short info, using keywords. The notification can be of any severity high scale, meetings, and new matches. | Performance |
| Dashboard | This component displays the UI to the users as soon as they log in. |  |

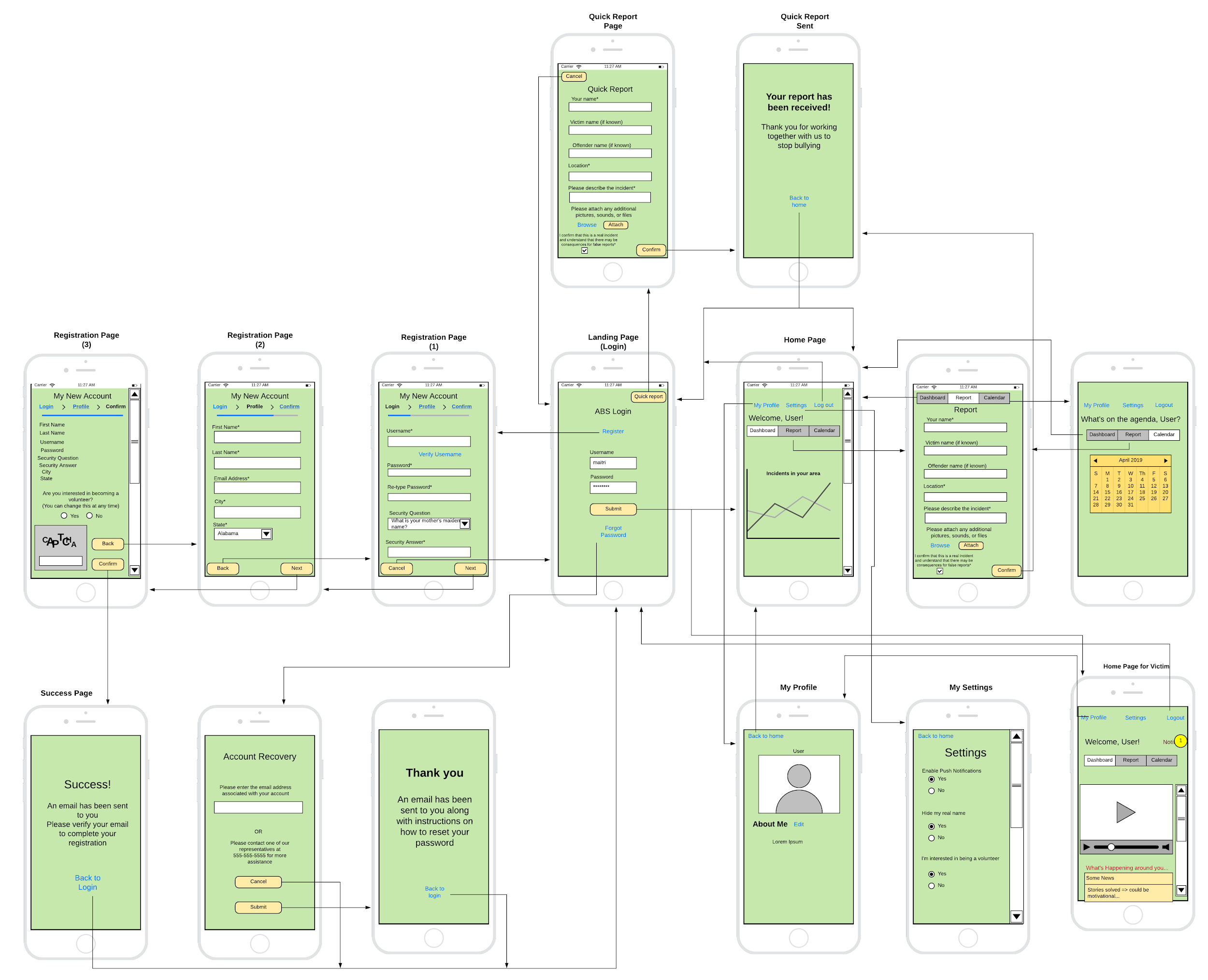


# USER EXPERIENCE / USER INTERFACE ARCHITECTURE

User interface and user experience can make or break a business. When considering these things, we have to make sure that accessibility and ease of access is ensured for all persons no matter what skill level with technology. In illustrating these ideas, a navigation flow and wireframe diagram serve as proper visuals.



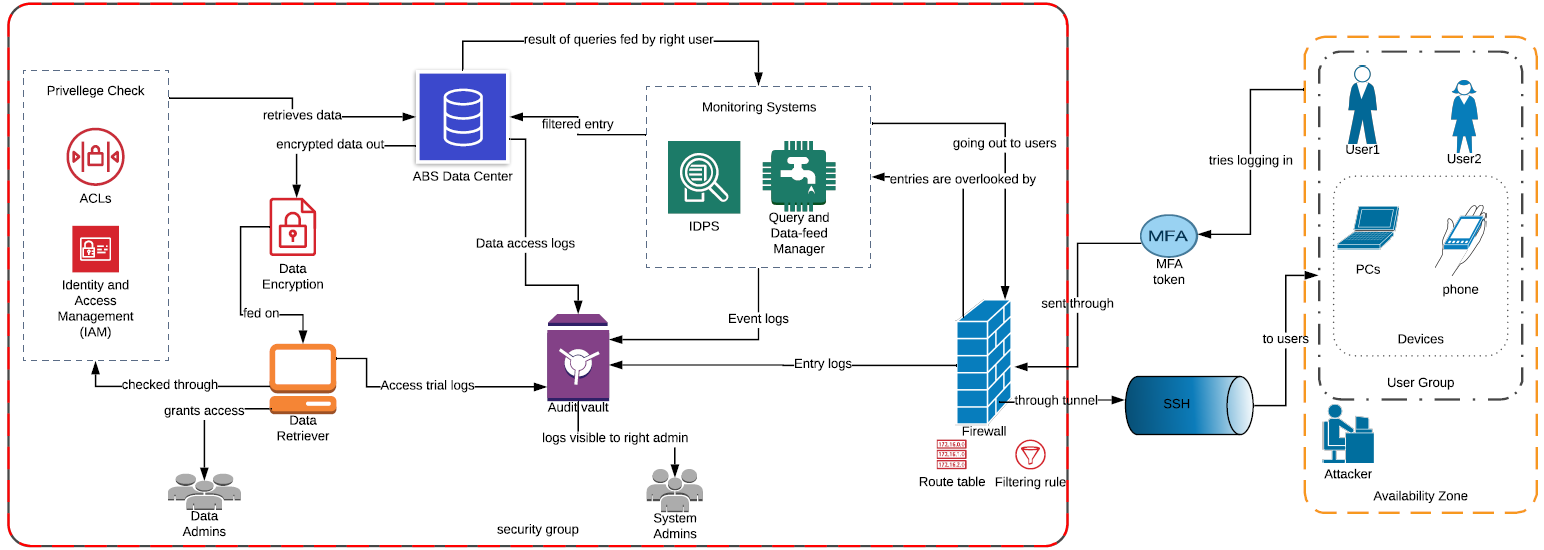
Navigation flow - The diagram above shows the simple process by which users can log in or register, as well as other login options. With this development, simplicity is key.



The wireframe is where the bulk of the information lies. From here, we can see the direction of each button and how information flows from one page to the next. When considering different types of views, (victim, volunteer, guest) we deduct that those views will mirror the one pictured in the diagram.

# SECURITY ARCHITECTURE

Security of systems and software is great issue in today’s world. It is hard and almost impossible for a system to exist with integrity, confidentiality and availability without security. System Architecture exist for tackling this issue. This architecture helps to integrate software and other systems to monitor the processes and activity of our system. It helps to predict and identify the loose points on our system and provides restriction and helps to develop patches to make it secure.



**Figure: System Security Architecture**

The figure below is the System Architecture Design of our Anti-Bully System. The architecture has components like firewalls, ACLs, IDPS, etc. All these components fall into some category that we have separated them into. They are all described below.

* **Availability Zone**

This describes to whom the system is visible and accessible to with right method. There are user group and also some threat agents like hackers who can view and use our website. The security is of a big concern because of these fake-users or threat agents or attackers who will try to manipulate or exploit our system to get the information that they don’t have right and privilege to view.

* + - **User Group**

This is a group of our users. The system users are Volunteers, Victims, Professional Helpers mostly but sometimes Guest Users are also going to use the system to report the on-going or bullies that they witnessed. They are going to any device that they are in contact with right now.

* + - **Devices**

These are basically all the devices that support our system. Our system will be compatible with any device that exists today. Some known examples of existing platforms are IOS, Mac, Android, Windows, Linux, etc.

* **SSH and MFA Tokens**

Multi-factor Authentication is a process of authenticating the users. This adds extra-layer of security on the user accounts. This system asks the user to input something they have, something they know and something they are as per Rouse, M. from Search Security[[1]](#footnote-0) to make the login and system more secure.

SSH is protocol called Secure Socket Shell which is used to connect one system to another through tunnel like secure structure for file and data transfer.

* **Security Group**

This is a group of software and security layers that helps maintain the integrity and confidentiality of the data stored in ABS data center.

* + - * **Firewall**

Firewall is the most general but effective and important security layer. It has protocols which helps limit the network traffic from and to the system. This layer deflects suspicious tries to exploits and attacks.

* + - * **Monitoring Systems**

This is combination of systems like Intrusion Detection and Protection System (IDPS) and Query and Data-feed Management systems. These systems help to detect the intrusions and any suspicious activity on the system. This system also alarms the admins in serious cases.

* + - * **ABS Data Center**

This is the Data Center that needs the protection and for which all the security layers are added. The data center is going to be cloud based (we are planning to use Tresorit Cloud Service for Data Storage).

* + - * **Data Retriever**

This is a device that Data Admins are going to use to retrieve the data that they need to work on. They will be allowed to view the data only if they have the privilege for it. They also need to log in using their credentials to use the device.

* + - * **Privilege Check**

This is where the privileges of the admins are going through trial. After the request to retrieve data from the data retriever device, it will go through Access Control Lists (ACLs) and Identity and Access Management (IAM) to make sure if they have the right and privilege to view and/or edit the data.

* + - * **Data Encryption**

The is a process that is used to encrypt the data that is retrieved from the Data Center. Without the encryption the data is visible to everyone after its retrieval. The right admin will have the decryption code which they use to decrypt it.

* + - * **Audit Vault**

This is a vault which has collects the logs from every other process which will be visible only to the right system admins. These logs will be helpful to detect the intrusions, system faults, etc. and to assess the breach which could help give explanation of the breach and ways to prevent the similar future breach.

* + - * **Admins**

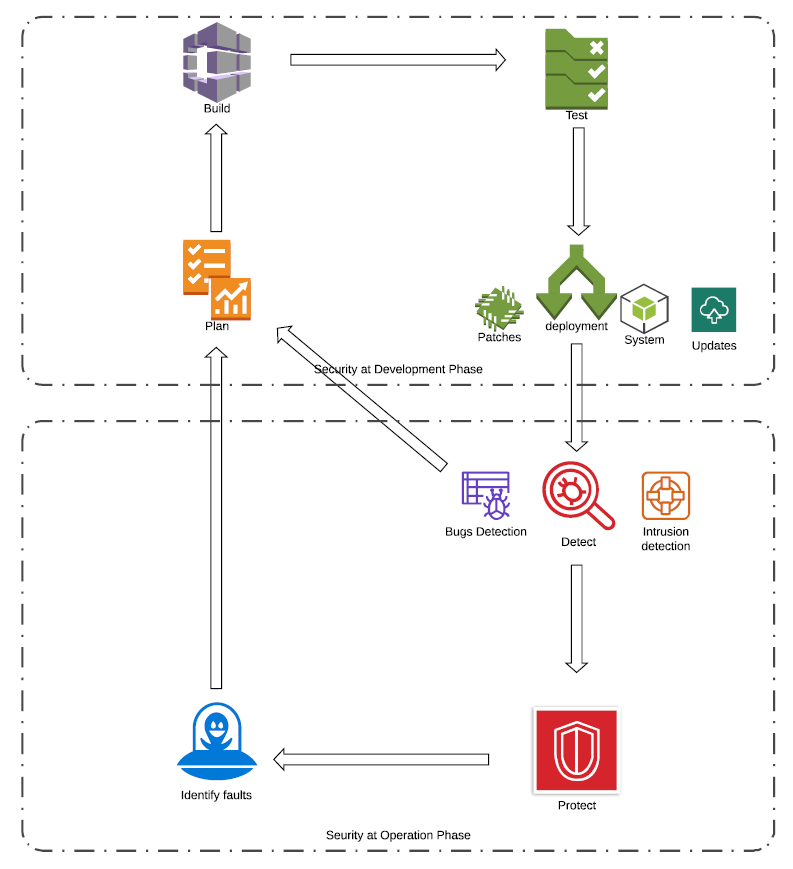
Even if they have the best software and processes to secure the systems, they will not be productive without the admins, therefore admins are the system overseers. They keep track of the use of system and data stores and ensure those security processes are working right.

*System Security Cycle*

Security is a process which never ends. The best way to make sure the system is secure is by implementing the idea of security right when we start building the code. The bugs in the system make the system most vulnerable. To ensure most vulnerable and crippling bugs are fixed, we need to build tests as we move on for the processes and codes we write. After the system has gone through lots of test, it is deployed for use and is available to the end-users.

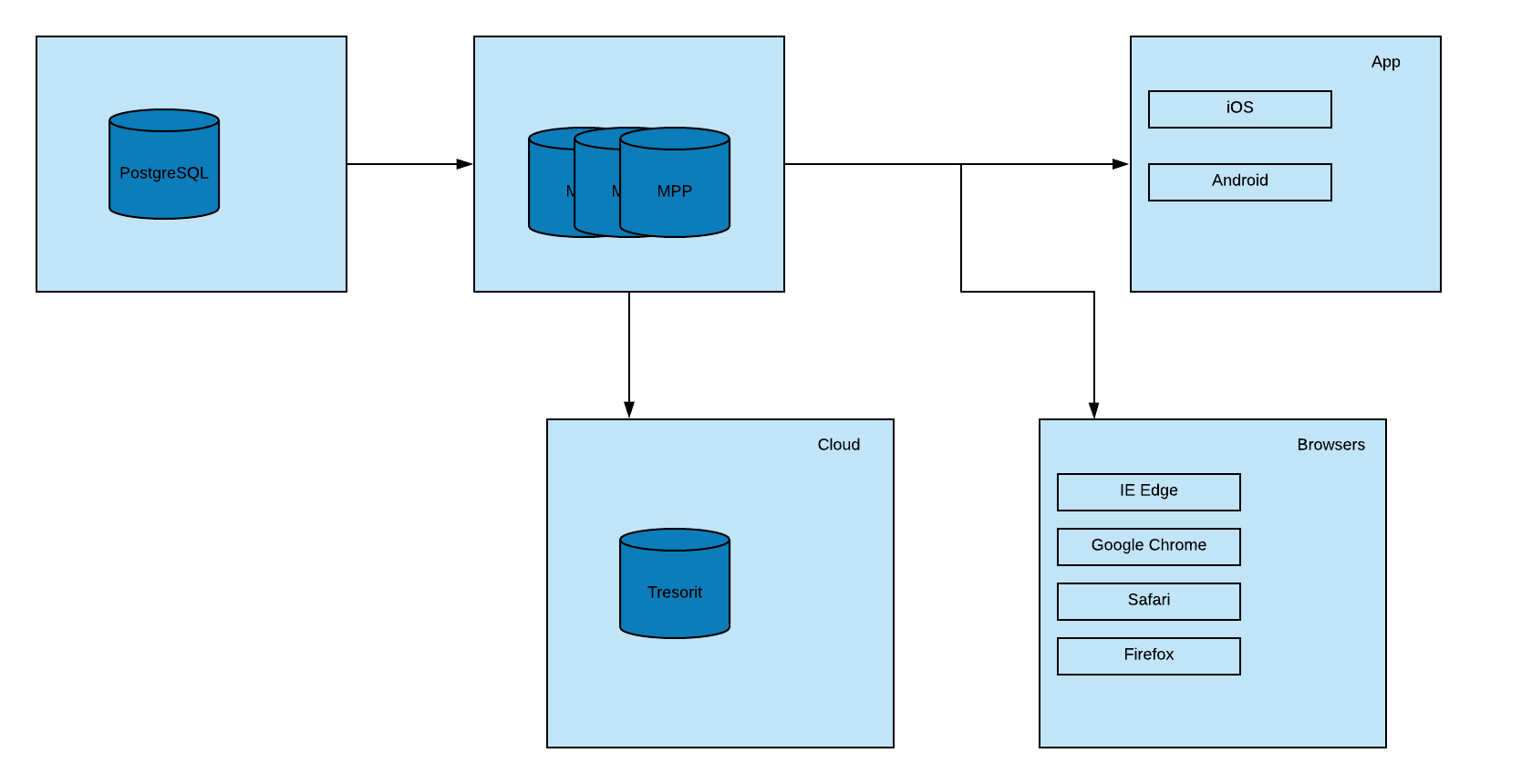
The security process doesn’t end here. While the system is in its operation phase, some new bugs are identified. Sometimes, the bugs that were invisible to developers and testers is visible to attackers which make the system vulnerable and will be exploitable. This is what we will need the above described processes to make sure the system is affected at the least. The protection of the system comes first after the attack and after the system is declared protected from current incident, we move on to identify the faults which led to a system breach. Planning to fix the fault will follow the identification of the fault. Then after, comes again building the fix and deployment of patches for the fixes.

This cycle keeps going. In case of identification of a new bug, it directly goes to planning phase and it deploys updates for the bug fixes. The figure below shows that a security is a cycle and never ends.

****

**Figure: Cycle of Security**

# INFRASTRUCTURE/DEPLOYMENT ARCHITECTURE



|  |  |
| --- | --- |
| Component | Description |
| PostgreSQL | A powerful open-source database that provides a cheap, effective solution. |
| Data Warehouse | It is IBM PureData for analytics, which is a MPP solution. The management node runs on RHEL 6. |
| Browsers | These are the major versions of browsers that pull in information. |
| Apps | These are the apps that pull in information. |
| Cloud Tresorit | This is the Cloud in which information will be stored |
|  |  |

# TECHNOLOGY STACK

The section explains and shows the use of different systems and software within the system we are building. The system will be deployed as a website and app. Every platform of operating system which are IOS, Mac, Windows, Android, Linux will support the system. The front-end of the system is going to be built with HTML, CSS, and/or JavaScript. The back-end is going to be built with Python and/or Java since they support the various frameworks which can be useful while building the website. The charge for using these components are free.

But, for storage of the data, we will use Tresorit cloud service which will cost $30 per month for users more than 10 and the capacity of the store starts with 1 TB. They have their own security policies and features integrated within the system which can be used by admins to monitor the system.

Development of the system is going to cost the most. According to article on FitSmallBusiness.com[[2]](#footnote-1), the cost of taking a professional help to build a website is going to range from $500 to $10000 plus depending on the professional choices and an extra cost of hosting will range from $50 to $250 a year. And after the system is built, the system will need regular maintenance which will cost 15% to 20% of total development cost as per FierceWireless.com[[3]](#footnote-2).

**Calculation of Cost:**

Development Cost: $7000 (approximation)

Cost of hosting a website: $100

Cost of Cloud Storage: $30 per month

Cost of Maintenance: 20% of Development Cost = $1400 per year

**Year0 = Cost of Development + Cost of Hosting a Website + Cost of Cloud storage per year**

**= $7000 + $100 + $30 \* 12**

**= $7460**

Cost in Year 0 is the cost computation of system when it’s at development phase.

**Year1 = Cost of Hosting a Website + Cost of Cloud storage per year + Cost of Maintenance**

**= $100 + $30 \* 12 + $1400**

**= $1860**

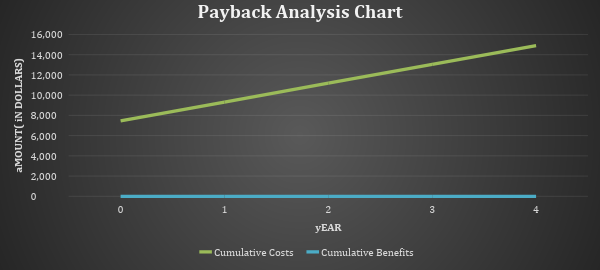
Cost in Year1 and above is the cost of system’s expenses after development.

The costs and expenses mentioned above are all approximation. The cost of the system is going to be same for some years and then it will rise gradually over time since at some period, the economical useful life will be over.

|  |  |  |  |
| --- | --- | --- | --- |
| **Architecture Component** | **Technology Choice** | **Justification** | **Cost** |
| Front-end | HTML, CSS, JavaScript | For building and polishing website. | No. |
| Back-end | Python, Java | For back-end development. | No. |
| Database System | PostgreSQL | It’s a database management system which is used to manage data stores. | Open Source |
| Cloud Data Store | Tresorit Cloud Service | A cloud storage service with all the security policies and features embedded. (Starting Capacity: 1TB) | $30 per month for over 10 users. |
| Monitoring System | Admin Center | It's a monitoring system that allows the admins to look over their data center. | Integrated within the Cloud |
| Supporting Devices | All Platforms | The system is going to be operated in all known platform. | N/A |

**Table: Technology Stack Use and Criteria**

The financial benefit of the system is proportional to what Maitri is going to charge the client organizations. If the Maitri is not charging for the use of its system, then the financial benefit is $0. But, since the Maitri is building the system to make our community a better place to live, the benefit of this system can not be measured only with financial perspective. It has some great intangible benefits. The graph below shows cost and benefit analysis of the ABS system. Obviously, we are assuming the Maitri organization have no financial benefits from this system.



**Figure: Payback Analysis Chart**

# Glossary

|  |  |
| --- | --- |
| Term | Description |
| NFR | Non-Functional Requirement(s) |
| OMS | Order Management System |
| ERD | Entity relationship diagram |
| POS | Point of Sale |
| CRM | Customer Relationship Management |
| EDI | Electronic Data Interchange |
| BI | Business Intelligence |

# References

Security Cycle

1. https://twitter.com/LMaccherone/status/843644744538427392
2. https://www.fulcrum.pro/newsletter/may-2018-newsletter/cybersecurity-framework/

Security Architecture

1. https://www.oracle.com/technetwork/database/security/wp-security-dbsec-gdpr-3073228.pdf
2. https://www.sejustloveit.com/2017/07/gdpr-oracle-max-data-security-architecture-se/

Technology Stack

1. https://db-engines.com/en/ranking\_osvsc
2. https://blog.capterra.com/free-database-software/
3. https://www.techradar.com/news/the-best-cloud-storage#best-business-cloud-storage
4. https://tresorit.com/security
5. https://fitsmallbusiness.com/website-costs/
6. <https://www.fiercewireless.com/developer/maintaining-app-critical-to-its-overall-success>

Infrastructure Architecture

1. <http://www3.cis.gsu.edu/dtruex/courses/MBA8125/Lectures-pdf/Day7-Architectures-pearlsonCh6.pdf>
2. <https://developers.google.com/apis-explorer/#p/>

1. Source: <https://searchsecurity.techtarget.com/definition/multifactor-authentication-MFA> [↑](#footnote-ref-0)
2. Reference: <https://fitsmallbusiness.com/website-costs/> [↑](#footnote-ref-1)
3. Reference: <https://www.fiercewireless.com/developer/maintaining-app-critical-to-its-overall-success> [↑](#footnote-ref-2)