

Design of an Interactive Geo-Location Mobile Application for Civil Societies in East Africa

Emil Patrick, Judith Leo, Shubi Kaijage

Department of Communication Science and Engineering, Nelson Mandela African Institution of Science and Technology, Arusha, Tanzania

Email: patricke@nm-aist.ac.tz, shubi.kaijage@nm-aist.ac.tz, judith.leo@nm-aist.ac.tz

How to cite this paper: Patrick, E., Leo, J. and Kaijage, S. (2021) Design of an Interactive Geo-Location Mobile Application for Civil Societies in East Africa. *Journal of Software Engineering and Applications*, **14**, 549-562.

<https://doi.org/10.4236/jsea.2021.1410032>

Received: April 30, 2021

Accepted: October 10, 2021

Published: October 13, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper reports the developed mobile application through the use of advanced technologies such as mobile computing, cloud computing and Global Positioning System (GPS) in order to solve the challenges. The purpose of this study is to find out how the recent advances and mass adoption of ICT by the public can best be leveraged to enhance the performance of Civil Society Organizations (CSOs) in East Africa. The data collection techniques used are through questionnaires, observations and conducting interviews with different stakeholders in the civil society arena *i.e.*, the donors, the CSOs and the people in society that are given a voice by civil societies. In the system development phase, agile development methodology was used. Analysis of the data collected showed that there is a gap in the availability of a single or a centralized platform which can be easily accessed, user-friendly and reliable where different actors can readily get reliable and up-to-date information about the available CSOs they are interested in. To address this problem, an interactive online directory of CSOs has been developed. The platform is mobile based and enables CSOs to register and fill up their current details, thus ensuring that there is always correct and updated information. The platform is equipped with, among other features, a geo-mapping facility which enables users of the system to correctly geo-locate their civil societies of interest on a map view. The results of system evaluation showed that 88.125% of users were satisfied with the system basing on the evaluation criteria.

Keywords

Civil Society Organizations, Mobile Computing, Design Science Research, Geo-Mapping

1. Introduction

Civil Societies are defined by the World Bank as the diverse groups of non-go-

vernmental and not-for-profit organizations that operate in public life, express the values and interests of their members and others, based on ethical, cultural, political, scientific, religious or philanthropic considerations [1]. Civil Society Organizations include but are not limited to labor unions, community groups, non-governmental organizations (NGOs), indigenous groups, charitable organizations, faith-based organizations, professional associations, and foundations. Civil societies can operate globally *i.e.* operating in multiple countries or they can be local to a particular country.

Examples of global CSOs are Amnesty International, World Wide Fund for Nature (WWF) and Save the Children. Local organizations examples are HakiElimu (Tanzania), Kenya AIDS NGOs Consortium (KANCO) and Golden Center for Women Rights (GCWR) in Uganda.

Civil Societies play an important role in society by acting as an unbiased tool for collective bargaining towards an inclusive, prosperous society. According to Klaus Schwab [2], some functions of Civil Societies are:

- **Watchdog:** They hold institutions to account and promote accountability and transparency;
- **Advocate:** They raise awareness of issues and challenges in society and advocate for change;
- **Provision of Services:** They help to deliver various services to meet needs in society in areas such as health, education, food and security. Also they implement disaster management, preparedness and responses in case of emergencies;
- **Provision of Expertise:** They help to bring unique and expert knowledge and experience in order to shape strategy and policy. They also identify and build solutions to societal problems;
- **Capacity Building:** They provide education, training and help build capacity in various areas;
- **Incubator:** They help in the development of solutions that may take a long time until payback is realized;
- **Representation:** They help the marginalized and under-represented get a voice;
- **Citizenship Champion:** They encourage engagement of citizens and support the rights of citizens;
- **Solidarity Supporter:** They promote fundamental and universal values;
- **Defining Standards:** They create norms that help to shape the market and the state activities.

Despite the important role played by CSOs in the society, by acting as an unbiased tool for collective bargaining towards an inclusiveness of marginalized groups and creation of prosperous society, much of the services offered do not easily reach the intended beneficiaries and stakeholders due to the fact that the CSOs still use paper-based and in-effective systems. For example, a CSO named Voice of Youth which is based in Arusha-Tanzania faces the challenge of finding/getting volunteers in their socio-economic development programs; because

of lack of effective mechanism for sharing information to the public about their activities.

The recent proliferation and mass adoption of ICTs by the public presents exciting new opportunities to CSOs to raise awareness about their work, find volunteers, connect with a wide range of individual and institutional donors and raise money. Some of CSOs have embraced ICTs by using technologies such as websites and social media accounts to widen their reach. However, there is still lack of a single online or centralized repository where accurate and updated information about CSOs can be found and browsed easily using modern technologies such as mobile phone applications. Hence, these challenges call for a need of ICT interventions. This is because ICT such as mobile applications have proven to be successful in areas such as health [3], agriculture [4] and m-learning [5]. Therefore, this study aims at bridging the gap by developing an interactive mobile application of CSO information which will be useful in assisting end-users such as communities and related stakeholders to easily access, browse and search information of CSOs in a single online or centralized repository.

2. Methods

The design science research methodology (DSRM) was used in this study. Design Science Research Methodology is a research activity that invents or builds new, innovative artifacts for solving problems or achieving improvements. DSRM creates new means for achieving some general goal, as its major research contributions. Such new and innovative artifacts create new reality, rather than explaining existing reality or helping to make sense of it [6]. DSRM is composed of three stages namely problem identification stage, solution design stage and evaluation and validation stage. DSRM methodology was preferred because it is outcome-based. It offers guidelines for evaluation and iteration within the project and focuses on improving the functional performance of the interfaces [7] which is the main goal of this implementation.

2.1. Area of Study

This study was done at the Eastern Africa National Networks of AIDS and Health Service Organization (EANNASO) in Arusha-Tanzania. In addition to EANNASO, further research was conducted in other CSOs located in Dar-Es-Salaam and Dodoma regions in Tanzania. The choice of these regions for the study is because they are major cities and have a substantial number of CSOs according to data from the Foundation for Civil society [8] as depicted in **Figure 1**.

2.2. Problem Identification Stage

During the stage of identifying the problem, a literature review was first conducted to understand the current situation on how ICT is leveraged by players in the non-profit sector *i.e.* CSOs, Community Groups, NGOs etc. in streamlining

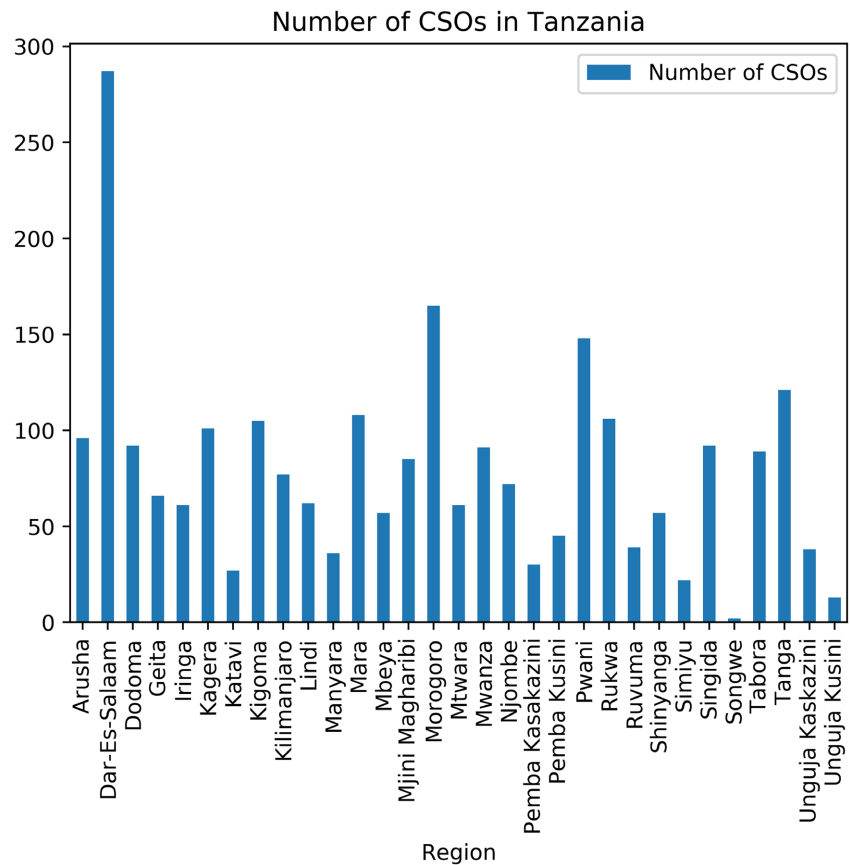


Figure 1. The number of CSOs in each region in Tanzania.

their operations and strategic objectives.

In the process, physical interviews, questionnaires and observations techniques with 140 players in the non-profit sector in Arusha, Dodoma and Dar-Es-Salaam regions were conducted. The participants comprised 72 males and 68 females. The participants had an average age of 35 whereby 45% of them came from CSOs, 35% from Community groups and 20% were individuals who are in the marginalized category such as the elderly, the disabled and unemployed youths.

In the interview sessions, questions such as 1) Have you ever wanted information about activities of a CSO/Community Group but did not know where to look? 2) Have you ever needed contacts of CSO/Community Group and failed to get one? 3) Have you ever obtained information about a CSO/Community Group but later found out that the information is outdated or inaccurate? 4) Have you ever tried to find physical location of CSO/Community but failed? 5) If you are a leader/owner of a CSO, have you ever wanted to publish/publicize your CSO and its activities but failed to find a suitable platform to do so? 6) In your opinion, how can we enhance accessibility to information about CSO/Community Groups? 7) Do you think a mobile application with information and location about CSO/Community Groups will help to solve some of the challenges you are facing? The feedback of the interviews was recorded and then, analyzed through the use of

inductive coding approach. The analyzed feedback were tabulated to show the frequency of responses from respondents in order to determine the most persistent problems facing the current landscape of CSO as far as information is concerned. This in turn would help the study in devising the solution to address the major pain points.

2.3. Solution Design and Development

Rapid Application Design (RAD) methodology has been used in the development of the solution to the problem identified in this study. RAD was used, because it offers iterative development with high quality result of information systems [9]. In the design and development process, the study conducted a total of five iterations in which features such as UI/UX design, application speed, scalability, security and robustness were evaluated and revisited as per user feedback on the prototypes. The prototypes were developed in Android/Java/Kotlin (frontend) and NodeJS/MongoDB (backend). NodeJs was chosen for the backend because of its high speed in serving API requests since it uses event-driven architecture and has non-blocking IO (Input/Output) [10]. MongoDB NoSQL database was chosen because it has more scalability and efficiency in storage and access of data than traditional relational databases like MySQL [11]

2.4. System Evaluation

In the system evaluation phase, a field study was conducted using the final prototype. The field study consisted of 60 diverse individuals who are players in the non-profit sector in Arusha, Dar-Es-Salaam and Dodoma regions. The participants were 36 males and 24 females whereby 55% were from CSO, 25% from Community Groups and 20% from marginalized groups (elderly, disabled, unemployed youths etc.). The questions asked in the evaluation study were on the assessment of the developed system if it has enhanced the access to information about CSO/Community Groups to interested stakeholders or not. Other questions focused on the ease of use, security, privacy and interoperability of the developed system and then, the results were processed using Pandas library of the python programming language.

3. Results

This section presents the results obtained towards the development of an Interactive geo-mapping online directory of CSOs in East Africa. The section is composed of three parts. The first part discusses the results obtained from the requirement collection of the research. The second part discusses the results of the system design and development phase. The third part discusses the results of the system evaluation process.

3.1. Proposed Functional and Non-Functional Requirements

The functional and non-functional requirements collected from the users for the

proposed system are summarized in **Table 1**. The requirement data was collected by using questionnaires, observations and conducting interviews with different stakeholders in the civil society arena.

3.2. Results of the System Design Process

The system has been designed following the client-server model. Based on the analysis of the requirements, the server (backend) has been designed following the Model View Controller (MVC) architecture because of the inherent benefits of this model such as enabling scalability, maintainability and robustness of software. Additionally, MVC makes reuse of code possible and enables different developers to develop the system in parallel. With MVC pattern, parts of the system can be modified without affecting the system as a whole [12].

The frontend has been developed in Android using Java/Kotlin following the Model View ViewModel (MVVM) pattern which is the recommended architecture for building robust, production-quality applications. MVVM architecture

Table 1. Requirements of the proposed system.

| Functional Requirements | | |
|-----------------------------|--|---|
| SN | Requirements | Description |
| 1 | Display up-to-date information about CSO/Community in an intuitive and interactive UI/UX | Users should be able to browse the directory of CSOs on their mobile devices in a well presented UI/UX |
| 2 | Collect and upload details of CSO/Community Group to the backend so that it can be displayed in the platform | CSO/Community Groups should have the ability to enter their information and send it to the system in order for it to be displayed in the platform for everyone to see |
| 3 | Search CSO per keyword | The system should enable users to search for data using keywords |
| 4 | Display CSO location on a map | Users should be able to locate the CSOs on a map, and be able to get directions on how to physically locate the CSOs |
| Non-Functional Requirements | | |
| 1 | Security/Authentication | The system should allow authenticated users to access its APIs. Also user passwords should be salted and hashed instead of being stored in plain text in the database |
| 2 | Privacy | The system should not share user data without express consent. In addition, the system should request for user's consent before collecting sensitive data such as a user's location |
| 3 | Accessibility | The system should be easy to use and access |
| 4 | Maintainability | The system should be easily maintainable |
| 5 | Scalability | The system should be designed in such a way that when users increases, the system can also be scaled up to accommodate the increased number of users. This has been achieved by reducing coupling between the different modules of the system |

implements the principle of *Separation of Concerns* between the different modules of the mobile application. Separation of concerns achieves modularization of the application making it easy to scale, read and unit-test the application [13]. **Figure 2** shows the architecture of the whole system *i.e.*, frontend (MVVM) and backend (MVC). In the frontend, the Activity/Fragment is responsible for presenting the UI to the user. The ViewModel is responsible for providing data to the UI. The Repository is a mediation layer that selects whether to pull data from the local database or making an API call to the remote data source (webservice). The SQLITE database is a local store of data on the mobile device. The transfer of data between the mobile frontend and the backend is accomplished using JSON (JavaScript Object Notation) format. In the backend the Controller is responsible for receiving and sending responses to API calls. The Model is a schema describing the structure of data and the View is a presentation layer that displays data to the user.

The following are the features of the proposed system

1) An android mobile application which will be responsible for providing a UI/UX for the user to interact with the system. The mobile application renders a view that displays all the CSOs in the system. It provides a form that users can use to upload information about CSOs. In addition, it senses the user's location (after requesting for permission from the user) and displays map view that shows all CSOs. The mobile application also provides search functionality.

2) A backend which is written in NodeJs using express framework. It is hosted in a cloud application hosting service called Digital Ocean. The backend implements a RESTful API that provides routes that serve HTTP requests from the mobile frontend in the JavaScript Object Notation (JSON) format.

3) A database for storing CSO information. The database used by the system is a MongoDB NOSQL document based database. This system uses MongoDB Atlas which is a database as a service (DbaaS) hosted in the cloud on Amazon Web

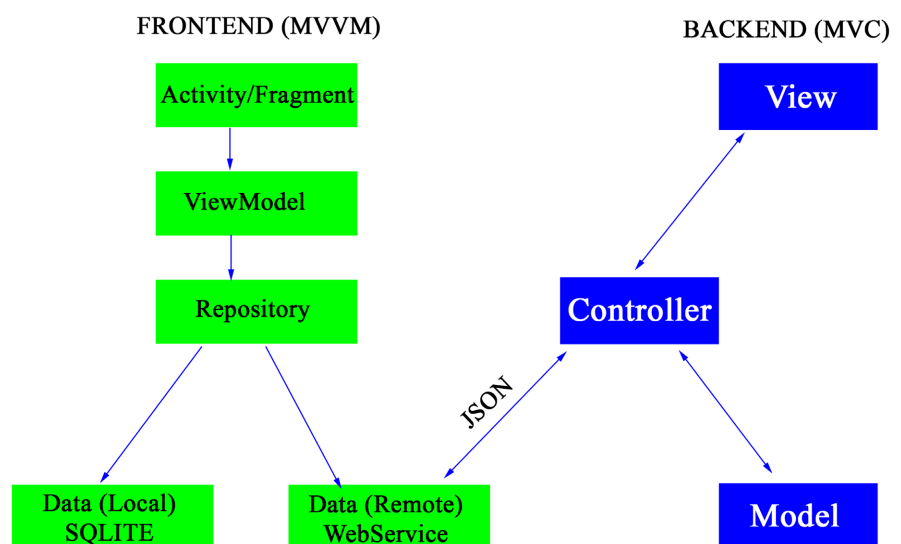


Figure 2. Proposed system architecture.

Service (AWS). Images and other media of the system have been hosted on Cloudinary platform which is a cloud service for hosting and transforming images and videos.

4) An administration panel that is web based which is developed using ReactJs frontend library. The ReactJs library has the benefit of saving time of development and minimizing the effort to maintain code [14]. The panel enables an administrator who has full rights of create, read, update and delete (CRUD) as well as to manage users and content of the system.

Figure 3 shows the conceptual framework of the developed system.

3.3. Use Case Modeling of the Application

Use case diagrams are used to show how one or more actors interact with the system. Use case diagrams depict what the system can do, *i.e.* how user inputs are handled in relation to the system's functional requirements [15]. In this application three actors were identified *i.e.* user, admin and Oauth Provider.

Users of the system interact with the mobile frontend. They can pull all data about CSOs, they can search data according to key queries such as country, region, CSO focus area etc. In addition, CSO/Community stakeholders can register to the system and upload information about their particular CSO by filling a form. The data is sent to the backend and stored securely in a cloud based MongoDB database via a NodeJs backend. In addition users can also geo-locate CSOs in an interactive Google map. The map also integrates Google's direction API that enables users to get estimated distance, time and directions to reach a particular geo-location [13]. There is an administrator who has all Create, Read, Update and Delete (CRUD) privileges of the backend. The admin manages users and content. Lastly there is an Oauth Provider who is an external entity that provides authentication services. For this application the authentication provider in use is Google's Oauth2 service. **Figure 4** shows the use-case diagram of the system.

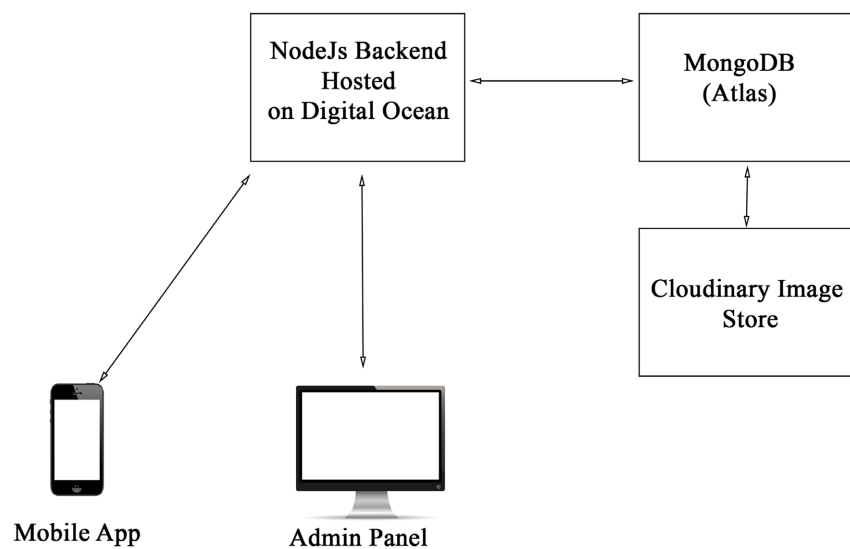


Figure 3. Conceptual framework of application.

3.4. Data Flow Diagram

A data flow diagram represents the processes, data stores, and external entities in a business or other system and the connecting data flows [16]. Data flow diagram can also be described as a graphical depiction of the flow of information in an information system. There are different levels of data flow diagrams depending on details. The first level is called context-level diagram which shows the whole system as an overview and how data flows between the different actors. The next level is called level 1 and contains more details. Subsequent levels are called level 2, 3 etc. each with more details than the previous level. **Figure 5** shows the context-level diagram of the CSO mobile application. Data flow from user to the system is authentication details for login or registration. In addition

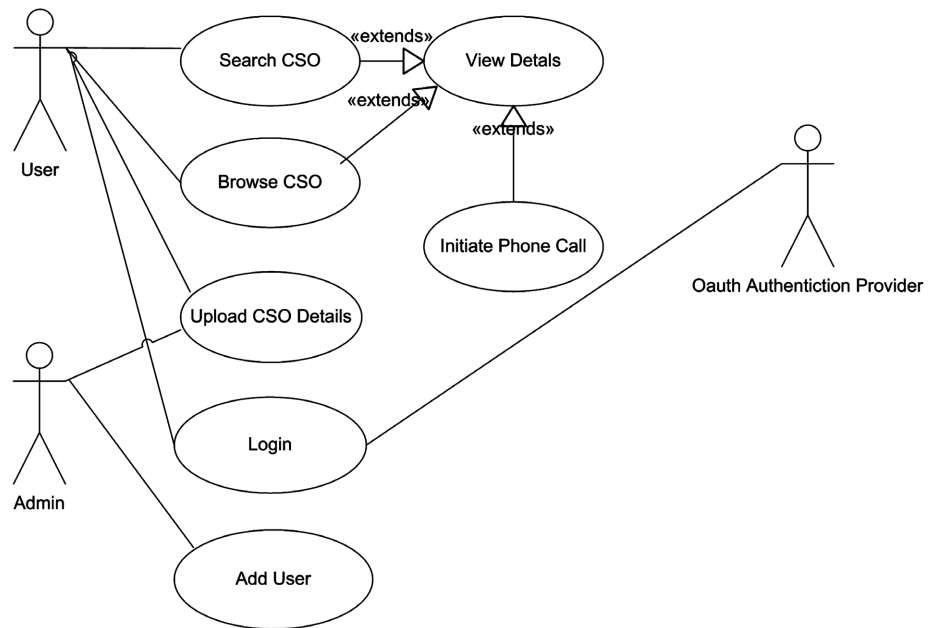


Figure 4. Use case diagram of developed system.

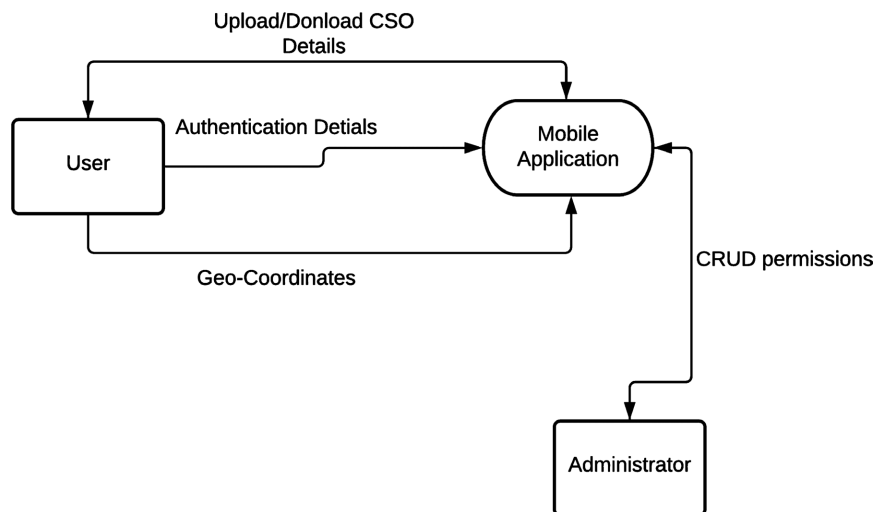


Figure 5. Context-level data flow diagram of the developed system.

user can upload their CSO details to the system. They can also browse and search CSO details from the system. Another data flow is from the administrator of the system. The admin has full CRUD permissions so they can send and retrieve data from the system.

3.5. System Security

The security of the system has also been taken into consideration. All communication between the frontend and backend is secured using Secure Hypertext Transfer Protocol (HTTPS). HTTPS is a protocol that is used to secure user data between a client and web server [17]. HTTPS is necessary to prevent phishing and man-in-the-middle attacks.

User passwords are stored in the database securely by using salting and hashing techniques. Passwords are not stored in plain text as this would compromise the system since any attacker who gains access to the database could read them. Instead passwords are salted in 10 rounds and hashed using bcrypt security nodejs module. Bcrypt is a scheme for hashing passwords. It is based on the Blowfish Block Cipher [18]. Bcrypt is resistant to brute-force attacks and greatly improves the security of a system [19].

User authentication is implemented using the JSON Web Token scheme (JWT). JWT is a standard used to create access tokens for an application [20]. The workflow of JWT begins with the client sending login or registration details to the REST API of the server. Once the server receives the username and password, it generates a token that certifies the user identity, and sends it to the client. The client stores this token and on subsequent requests, the client will append the token with the request back to the server. On receiving the request with the correct appended token, the server will know that the request comes from an authenticated client and not a fake client. In this application the JWT token is stored on the android mobile application using Jetpack Data Store [13]

3.6. System Requirement Evaluation

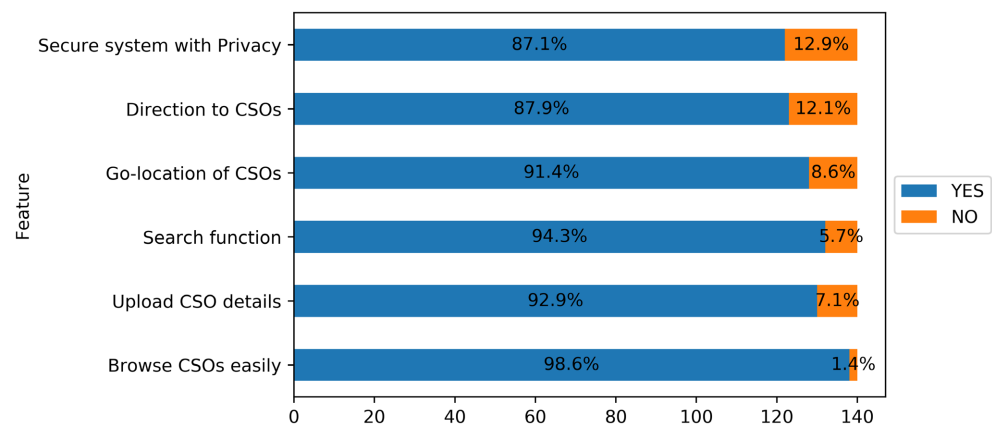
Table 2 describes how the system requirements have been fulfilled by the designed system

3.7. Overall System Evaluation

Figure 6 shows the overall results of system evaluation. The system was evaluated by giving a number of users questionnaires with a set of questions and tabulating the frequencies of their answers. In the evaluation process a total of 60 users; whereby 48 users were from CSOs and community groups and 12 from marginalized communities answered a set of eight questions about the system. The results show that on average 88.125% of users granted positive reviews of the developed system in accordance to different criteria such as its ease of use (usability), accessibility, security, speed, availability and enhancement of information availability of CSOs. The remaining average of 11.875% represents users

Table 2. System requirements evaluation.

| Requirement | System Verification |
|---|--|
| The system developed must have the ability to display up-to-date information about CSO/Community in an intuitive and interactive UI/UX | The developed system displays the CSOs stored in the database on a recyclerview UI element. Recyclerview is view with good aesthetic UI and is memory efficient as it automatically recycles views |
| The developed system must be able to collect and upload details of CSO/Community Group to the backend so that it can be displayed in the platform | The system has a form UI element that enables the user to fill in information about their CSO and send the data to the backend and database. The frontend uses Retrofit networking library to communicate with the backend. |
| The developed system must be able to display CSOs location in a map. | The system has integrated Google Map functionality that displays the CSOs as markers on a map. An info window displays the information of each CSO on clicking it. In addition, the user can get direction to CSO of choice; this functionality leverages Google's direction API. |
| The developed system should secure user data effectively | This is achieved by installing an SSL certificate on the server. The certificate enables encryption of all HTTP communication between client and server using HTTPS protocol. In addition, all passwords are not stored as plain text. They are salted and hashed using bcrypt blowfish algorithm then stored in the database. Some APIs are also protected using tokens and only accessible after a user is successfully authenticated. |
| The developed system should be simple, scalable and maintainable | The developed system has achieved these requirements by adopting the MVVM and MVC models in the frontend and backend respectively. These paradigms enhance code readability and modularization, making the application easy to maintain and to grow as users increase. |

**Figure 6.** Results of system evaluation.

who were unsure of some of the meaning of the concepts like security, availability and accessibility. Hence, the study received less numbers of users who were not satisfied with the system's performance. In some of the issues clarification and training were carried out in order to assist the unsatisfied users with proper and clear explanation or information.

3.8. Application Software Testing

Software testing is a method of finding errors in an application or program. It ensures that the application functions according to the requirements [21]. There are three kinds of tests namely unit testing, system testing and integration testing. Unit testing is done at the unit level while integration testing tests the integrated units of a module. System testing is done to test the overall functionality

of the system.

In the development of this application unit testing was done automatically to each unit by scripts generated by JUnit available on Android Studio [22].

3.9. Some Screenshots of the Application

User interfaces provide the first point of interaction between the users and the system. They are an important component of an application because they give the first impression of what the app can do to the user. **Figure 7** and **Figure 8** show some screenshots from the developed application.

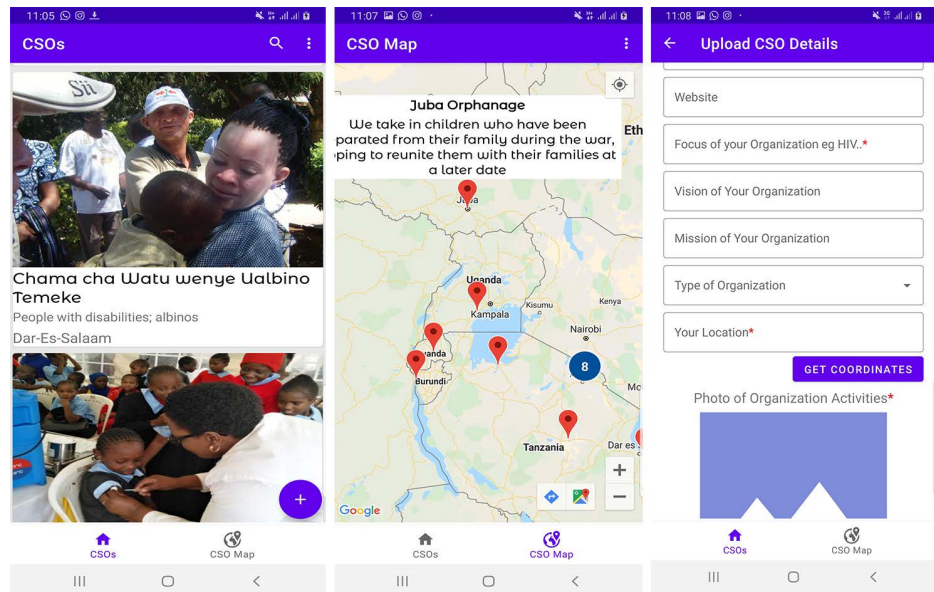


Figure 7. Some UIs of developed application.

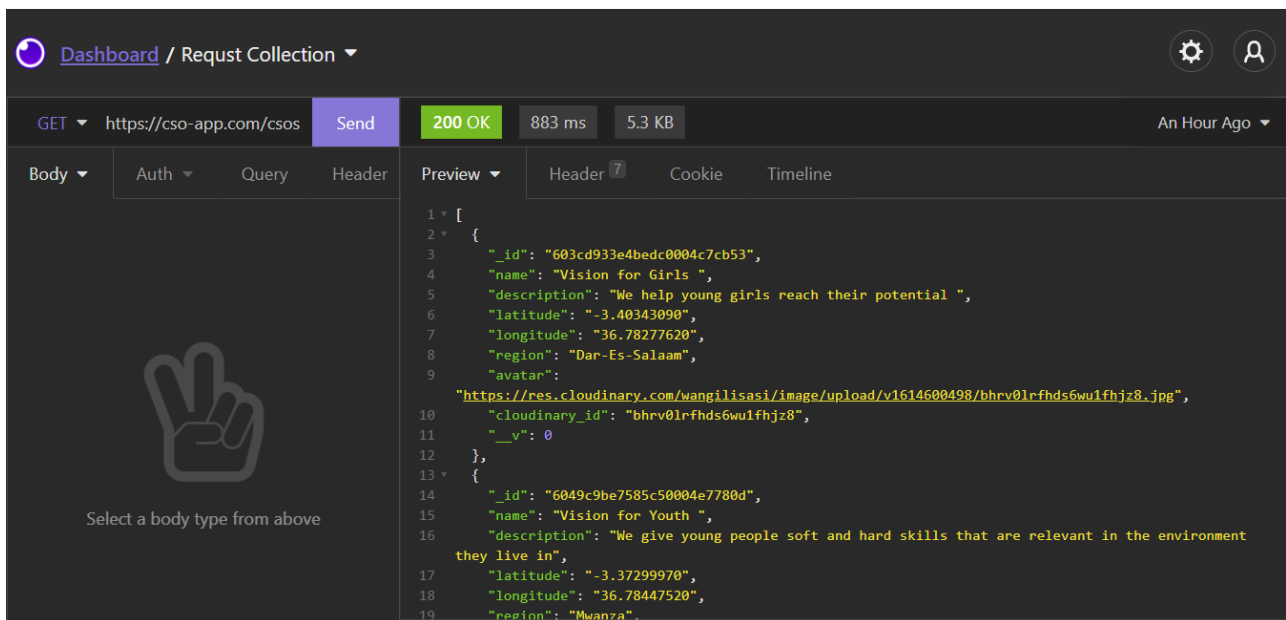


Figure 8. GET response from backend.

4. Conclusion

The non-profit sector is an important sector in society as explained in the introduction section of this study. Therefore people need to have awareness and easy access to CSO and community groups by using state of the art technologies like mobile applications that leverage inbuilt sensors of mobile devices. As part of future work the authors will work on developing this application further and equipping it with more features. Also the authors will work on developing a similar application for IOS users who use iPhones in order to widen the reach of the application.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] The World Bank (2020) Civil Society. <https://www.worldbank.org/en/about/partners/civil-society/overview>
- [2] Schwab, K. (2012) The Future of Civil Society. http://www3.weforum.org/docs/WEF_FutureRoleCivilSociety_Report_2013.pdf
- [3] Mramba, B.P. and Kajjage, S.F. (2018) Design of an Interactive Mobile Application for Maternal, Neonatal and Infant Care Support for Tanzania. *Journal of Software Engineering and Applications*, **11**, 569-584. <https://doi.org/10.4236/jsea.2018.1112034>
- [4] Donovan, K. (2017) Anytime, Anywhere: Mobile Devices and Services and Their Impact on Agriculture and Rural Development. In: World Bank, Ed., *ICT in Agriculture (Updated Edition): Connecting Smallholders to Knowledge, Networks, and Institutions*, 49-70. https://doi.org/10.1596/978-1-4648-1002-2_Module3
- [5] Crompton, H. and Burke, D. (2018) The Use of Mobile Learning in Higher Education: A Systematic Review. *Computers & Education*, **123**, 53-64. <https://doi.org/10.1016/j.compedu.2018.04.007>
- [6] Pries-Heje, J., Baskerville, R. and Venable, J.R. (2008) Strategies for Design Science Research Evaluation. *16th European Conference on Information Systems, ECIS*, Galway, August 2008, 3-6.
- [7] Geerts, G.A. (2011) A Design Science Research Methodology and Its Application to Accounting Information Systems Research. *International Journal of Accounting Information Systems*, **12**, 142-151. <https://doi.org/10.1016/j.accinf.2011.02.004>
- [8] The Foundation for Civil Society (2019) Directory of Civil Society Organizations 2017-2018. <https://thefoundation.or.tz/>
- [9] Lank, E., Withee, K., Schile, L. and Parker, T. (2005) User Centred Rapid Application Development. In: Guelfi, N. and Savidis, A., Eds., *Lecture Notes in Computer Science*, Springer, Berlin, 34-49. https://doi.org/10.1007/11751113_4
- [10] Bangare, S., Gupta, S., Dalal, M. and Inamdar, A. (2016) Using Node.js to Build High Speed and Scalable Backend Database Server. *National Conference of Advanced Web Technologies*, New Delhi, 19 March 2016, 61-64.
- [11] Chauhan, D. and Bansal, K. (2017) Using the Advantages of NOSQL: A Case Study

- on MongoDB. *International Journal on Recent and Innovation Trends in Computing and Communication*, **5**, 90-93.
- [12] Leff, A. and Rayfield, J.T. (2001) Web-Application Development Using the Model-View-Controller Design Pattern. *Fifth Proceedings of International Enterprise Distributed Object Computing Conference*, Seattle, 4-7 September 2001, 118-127.
- [13] Google Developer Guide (2020) Guide to App Architecture. <https://developer.android.com/jetpack/guide>
- [14] Le, T.A.V. and Tran, N.T.A. (2020) Implementation of React-Redux in Web Applications. *Journal of Software Engineering and Applications*, **6**, 4-12.
- [15] Bittner, K. (2003) Use Case Modeling. Addison-Wesley Professional, New York.
- [16] Aleryani, A.Y. (2016) Comparative Study between Data Flow Diagram and Use Case Diagram. *International Journal of Scientific and Research Publications*, **6**, 124-126.
- [17] Goodrich, M.T. and Tamassia, R. (2011) Introduction to Computer Security. 2nd Edition, Pearson, Waterloo.
- [18] Alabaichi, A., Ahmad, F. and Mahmood, R. (2013) Study on Informatics and Computer Applications. *Journal of Informatics & Applications*, **6**, 12-18.
- [19] Malvoni, K. and Knezovic, J. (2014) Are Your Passwords Safe: Energy-Efficient Bcrypt Cracking with Low-Cost Parallel Hardware. *Journal of Informatics Security & Applications*, **4**, 1-7.
- [20] Jones, M., Campbell, B. and Mortimore, C. (2015) JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants. *International Journal of Computer Security Applications*, **6**, 4-5. <https://doi.org/10.17487/RFC7523>
- [21] Everett, G.D. and McLeod, R. (2007) Software Testing. *Journal of Software Engineering and its Applications*, **7**, 17-21.
- [22] Google Developer Guide (2020) DataStore. https://developer.android.com/topic/libraries/architecture/datastore?gclid=Cj0KCOjwse-DBhC7ARIsAI8YcWL9YCJBBpUt1-ewTl3r484eId_qLLus6BvFrykFuO5NNPpE-Uhxz9UaAoZEEALw_wcB&gclid=aw.ds