



UNIVERSITY OF CAPE TOWN



DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Project Final Paper 2023

Title: Re-engineering a Platform for Managing Early
Childhood Development Resources

Author: Lwazi Sibeko

Project Abbreviation: BBP2

Supervisor(s): Melissa Densmore

Category	Min	Max	Chosen
Requirement Analysis and Design	0	20	20
Theoretical Analysis	0	25	0
Experiment Design and Execution	0	20	0
System Development and Implementation	0	20	20
Results, Findings and Conclusions	10	20	10
Aim Formulation and Background Work	10	15	10
Quality of Paper Writing and Presentation	10		10
Quality of Deliverables	10		10
<u>Overall General Project Evaluation</u> (<i>this section allowed only with motivation letter from supervisor</i>)	0	10	0
Total marks		80	

Re-engineering a Platform for Managing Early Childhood Development Resources

Lwazi Sibeko
SBKLWA003@myuct.ac.za
University of Cape Town
Cape Town, South Africa

ABSTRACT

This paper outlines a collaborative research project between the University of Cape Town Honours students and the Bhabhisana Baby Project, focusing on the refactoring of the 'You and Your Baby' content delivery system for the organisation's beneficiaries. We discuss the co-design and software development processes followed in order to achieve this, as well as analysing the portability and viability of the system in a diverse yet similar context. The results of this effort include the creation of a new system built with the Django framework. Usability tests demonstrated improved user experiences with the system. Additionally, co-design engagements with a different community, Sweetwaters, highlighted the system's adaptability and user-focused approach, showing its potential for being adopted by other non-government organisations.

CCS CONCEPTS

• **Human Computer Interaction** → *HCI design and evaluation methods*; • **Software and its engineering** → Software creation and management.

KEYWORDS

Human Computer Interaction, Web Development

1 INTRODUCTION

Through digital technology, non-governmental organisations (NGOs) can operate at higher levels of functionality. They can work more effectively from within the organisation by using it as an efficient tool to facilitate their work processes and increase their reach to supporters and beneficiaries [9]. Sustainability in a software context can either relate to creating software that takes into consideration the human and financial resources available from an organisation, or it can relate to the maintainability of the software system whilst it is in use by an organisation [10]. Both of these interpretations are relevant to building sustainable software for NGOs. This is due to NGOs' inherent resource constraints, which limit their ability to fund the creation and maintenance of software systems [3]. This challenge, therefore, affects the sustainability of software systems built by or for NGOs. Sometimes, NGOs have partnerships with universities, where they collaborate on a research initiative related to their work and shared aims [8]. Such initiatives include university students contributing resources to support the NGO's goals.

This project outlines a collaborative effort between University of Cape Town (UCT) Honours students and the Bhabhisana Baby Project to co-design You and Your Baby, a content delivery system for the NGO. This paper outlines the challenges encountered in the 2022 deployment of the You and Your Baby system and how these challenges formed the motivations behind the decision to

refactor the system. As a result, a new system was created to align with the previous and emerging functional and non-functional system requirements. The new system is built with the Django framework, leveraging its Model-View-Template (MVT) pattern, an Object-Relational Mapping (ORM) layer and built-in tools, enhancing the system architecture. Throughout this paper, we provide a comprehensive account of our co-design and software development processes, outlining the steps taken to do so. Additionally, we present our findings and engage in a thoughtful discussion of the project's progression.

1.1 Project Aims

The main objective of this project is to perform a system refactoring of You and Your Baby for the Bhabhisana Baby Project. This involves an analysis of the existing project architecture, identification of shortcomings, and the re-engineering of the current platform. In addition to this, our project aims to explore the system's performance and adaptability within similar contexts. By leveraging insights gained from our co-design engagements with different communities, notably the Bhabhisana Baby Project and Sweetwaters, we aspired to develop a user-centric system that can integrate into diverse contexts and cater to the specific requirements of various NGOs. This project aims to re-engineer the system into a more versatile and reliable solution, aligning with the evolving needs of the Bhabhisana Baby Project and potential users in similar contexts.

2 BACKGROUND

The Bhabhisana Baby Project (BBP) is a non-governmental organisation (NGO) based in Cape Town that aims to help caregivers from low-income communities care for their children with developmental issues. Established in August 2015, the organisation comprises an interdisciplinary team of eight members, each of whom is either a speech therapist, an occupational therapist, or a physiotherapist. They aim to address the health of infants from underserved communities within the first 1000 days after birth by providing early intervention programmes, including therapy, care training and guidance until caregivers can access the necessary support from the public health sector. Caregivers are typically enrolled in a programme for as long as it takes until they are fully integrated into and supported by the public health system. Caregivers receive a monthly 45-minute therapy session at their Athlone, Cape Town offices. These sessions usually involve providing caregivers with guidance on movement, play, stimulating language development, and feeding. In these sessions, caregivers receive advice on maximising developmental input by continuing the therapy sessions at home. Caregivers are expected to retain and remember the guidance given during sessions to implement at home.

You and Your Baby is a content delivery system resulting from the ongoing co-design effort between the CoMaCH, BBP, 3 UCT Honours students and a UCT Masters student. The system aimed to allow BBP beneficiaries to access resources and advice on caring for their baby between therapy sessions. You and Your Baby provides caregivers with curated content to help and educate them on navigating, taking care of and raising their baby.

3 ARCHITECTURE

3.1 2022 System Architecture

The previous system, developed by Swanepoel [12], was developed using WordPress. WordPress is a free and open-source PHP-based Content Management System that allows users to create, manage and publish digital content such as text, images and videos. The system relied on a MySQL database to store data. The Pods plugin was used to configure custom meta-boxes, which allowed for the customisation of content types and relationships between content types. Oxygen Builder is a website design and development tool used to create and customise the website's web pages. Custom post types and custom fields were used to manage content, and REST API endpoints were designed for data retrieval for the mobile application. HTML, PHP, and CSS were used to customise the user interface further and implement functionality. The mobile application was developed using Android Studio, with Kotlin for backend functionality and XML for frontend styling. The development approach of the previous system was to abstract the system into a staff-facing web portal, a user-facing web application and a mobile application for content access. The key features of the system were:

- The staff-facing web portal performed the functionality of allowing web portal users to create and manage content and users, emphasising content assignment functionality.
- The user-facing web application allows users to authenticate, view content based on categories and languages, and access assigned content. The mobile application was designed to have responsiveness, usability, and coherent branding features.

The problems identified with the 2022 deployment:

- The staff reported that it was difficult to use the web portal due to its unintuitive interface. Although they have been using the application for as long as they have, they still struggle to navigate it. One staff member stated it was "*tedious and time-consuming*" to upload videos and "*I usually put aside a whole day just to upload content*".
- The quality of JSON response returned by the WordPress API endpoints was large and poorly structured due to the lack of customisability that WordPress allows for API design. String manipulation needed to be done by the mobile application to extract file links.
- There was a lack of proper user authentication. Users were only authenticated based on whether the username existed. When content was retrieved for the mobile application, only the user ID was used. There was no password check or session validation performed. Therefore, a user's identity was not verified beyond username existence. As this platform

deals with retrieving sensitive assigned content, this could lead to users accessing content not intended for them.

3.2 Django Project Structure

Django's project directory structure consists of apps. An app is a self-contained module that serves a specific purpose within a web application [1]. It allows code to be developed into modular, reusable pieces that encapsulate functionality. The architecture of the new system is divided into four apps:

yayb2: This app contains the settings and middleware configurations of the system. The system's name is yayb2, which stands for You and Your Baby 2; therefore, following Django framework naming conventions, this core app's name is named after the system.

main: This app serves the functionality of providing the system with the primary function of the system. To display content to the users.

android: This app contains all the API functionality created to allow the mobile application to communicate with the system.

users: This app contains all the functionality providing the system with user management.

The Django framework follows a Model-View-Template (MVT) design pattern for each app, similar to the commonly known Model-View-Controller (MVC) design pattern.

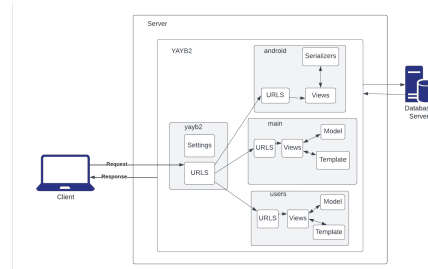


Figure 1: Architecture of the final system

Models define the structure and organisation of the data. Each model represents a specific data entity, such as a user and a content item.

Views are Python functions that receive web requests and return HTTP responses. They are responsible for processing user requests, querying databases, and returning the appropriate response

Templates define the layout and appearance of your web pages through the use of HyperText Markup Language (HTML) pages.

URL patterns determine how users can navigate to different application parts. URLs are mapped to specific views, allowing users to navigate to different pages.

When a client sends an HTTP request, the root URL routing system in the yayb2 app determines which app is responsible for handling this request and passes it onto that app's URL routing system. That app's URL router determines the view function should handle the request. The selected view function processes the request, generates an HTTP response (an HTML page), and sends it directly back to the client's browser. Each individual app defines its views, and it responds directly to client requests. This separation of concerns allows for modularity and scalability in Django applications. A view is responsible for retrieving the relevant content from

the Model. The View is also responsible for retrieving the relevant HTML Template necessary to render that web page.

3.3 Sweetwaters' Application Architecture

The functionality required for the Sweetwaters workshop deployment is accessing and viewing content. Therefore, the system deployed in Sweetwaters consists of just the `yayb2` and the main apps.

3.3.1 main The main app contains all of the code relating to the basic functioning of the web application. All of the logic relating to the users utilising the application by viewing content lies here. The templates directory contains all the HTML templates for the web pages rendered by this app. The app's views file includes the following views:

main: This view is responsible for rendering the main template of the website with all of the public content available in the database (shown at the bottom of the page).

content: This view is responsible for rendering the page for a single content item and the item from the database. This view accepts a `contentID` parameter, which fetches the file from the database.

category: This view is responsible for rendering a page for all the content items belonging to a single category and the items from the database. This view accepts a `category` parameter used to fetch all the public files from the database that match this category.

teesandcees: This view returns and renders the terms and conditions page.

Each view has its own URL directory leading to it in the main app's `urls.py` file.

3.3.2 yayb2 The `yayb2` app contains all of the primary settings and configurations for the entire project, serving as the project's entry point, as illustrated in Figure 1. It includes a `urls.py` file that routes all URL requests to their respective apps. Unlike other apps, '`yayb2`' does not have a '`views`' file. This app contains a settings file with the configurations necessary to control the project's behaviour. This file specifies database settings, defines password validation rules, and manages middleware for security and request processing. It also lists installed apps, both project-specific and Django's built-in ones, and sets URL routes for media, static files, and login directories. This central configuration file plays a crucial role in defining how the Django application functions and interacts with users.

3.4 Final Architecture

The source code for the final system is listed under Github Repository. The final architecture is built on top of the Sweetwaters' deployment architecture, which consists of main and `yayb2`. It also includes the additional functionality provided by the users and android apps. While the `yayb2` app is not different, the main app has some changes for the final implementation.

3.4.1 main The Installed Apps configured here now include the configuration for the Django Rest Framework and `jazzmin`, a theme to enhance the interface of the admin dashboard. Another change is in the database configuration, now set to connect to a PostgreSQL database. The currently deployed version on the UCT network configures the database to connect to the PostgreSQL database Docker container.

3.4.2 android The android app contains all the code relating to the Django REST Framework (DRF) API created to allow the system to communicate with the android app. The views file of this app includes all the API endpoints that allow the mobile application client to interact with the system. The API endpoints available are as follows:

android_login: This API endpoint allows the mobile application client to log in using a username and password. If successful, an authentication token is returned, which the mobile client will continue using to make future requests.

android_logout: This API endpoint terminates the logged-in session of the mobile client and deletes the authorisation token previously given.

get_all_content: The API endpoint allows the mobile client to get a JSON response of all the public content in the database.

get_assigned_content: Using the authentication token for validation, this API endpoint returns content assigned to a user (associated with the authorisation token).

Each endpoint has its own URL directory leading to it in the android app's `urls.py` file. This app also includes a `serializers` file that defines how the user and content data from the database must be serialized and converted to the JSON type returned to the mobile client.

3.4.3 Users app The users app contains all of the logic necessary to enable users to access assigned content. Its views file has an '`assigned`' view. This view is responsible for rendering the assigned template of the website, which displays all of the content assigned to a particular user. This view has a '`@login_view`' decorator, which ensures that a user must be logged in and authenticated first to access this view. This view has its own URL directory leading to it in the users app's `urls.py` file.

3.4.4 User Interface The user interface of the system with which the users engage was created using HTML templates. For the different pages, there are respective HTML templates. For styling, Bootstrap was used. Bootstrap supports mobile-first design [4], which is ideal, as we expect that the web application users will access the application with their phones. JavaScript scripts were used for the dynamic behaviour of filtering between content.

3.4.5 Web Portal To create the web portal, Django's built-in administration interface was utilised. The built-in administration interface simplifies the management and manipulation of data in the web application. It allows administrators to manage application data without interacting with the database. The benefits of using this interface include:

An automatic interface: The Django admin creates a basic user interface based on the defined models.

CRUD Operations: This admin interface allows you to perform Create, Read, Update and Delete actions on the database items.

Customisation: The primary automatic admin interface can be further customised to the developer's liking. For this project, `jazzmin` was used to customise the user interface.

User authentication: The Django admin uses the Django user authentication and authorisation, which ensures secure and authorised access to the admin portal.

For the Content model to be accessible through the admin interface, it was registered as a model in the '`admin.py`' file. Because

Django predefined the User model leveraged by the system, it is automatically accessible through the admin interface. An admin superuser account was created. This superuser has complete access to the system and can make changes and grant access permissions to other Users.

3.5 Authentication and Security

As this content delivery system needs to allow different users to access and manage various types of content, user authentication is a vital function. To access assigned content, users have to log into the system. Django decorators are used to require authentication to access certain views. The Django permissions system is leveraged to grant user roles, allowing selective control over which users can perform which actions within the application. This feature allows website administrators to be the only ones given administrative privileges to the web portal. Django's built-in User model handles user authentication processes. The login and logout views are predefined by Django, ensuring seamless execution of these operations' logic. Django's Object-Relational Mapping (ORM) system safeguards the system against SQL injections using parameterised queries and sanitising inputs.

For external access to the system via the API, Django authentication verifies credentials and issues tokens upon login. Token-based authentication adds an extra layer of security, ensuring only authenticated users with valid tokens can access endpoints. Permission classes control access to authenticated views, increasing overall system security and functionality.

3.6 Model and Database Design

For data storage, the system makes use of a relational PostgreSQL database. We leveraged the Django framework's default Object-Relational Mapping (ORM) layer to facilitate interactions between the system and the database. The Django ORM creates an abstraction layer between the system and the database. It allows the system to interact with the database using Python classes instead of using SQL statements. The Django ORM layer uses the models defined by Python classes and abstracts them into database tables. Therefore, the creation of the database tables is based on the models defined in the system, and thereafter, the Django ORM layer handles the responsibility of creating and modifying database tables.

The only model created for the system is the Content model in the main app. This model defines how content items are stored. The Content model has an assignedUsers property that is a Many-To-Many field that assigns users to content items. The User model utilised to represent User accounts comes from the Django Authentication package. The Django framework also makes use of other built-in models, as shown in figure Model structure, that are integral components of Django's authentication, authorisation, and administrative functionality, streamlining the development process and promoting secure and efficient web application creation.

3.7 Current Hosting Configuration

The system can be accessed from <http://bbp-1.cs.uct.ac.za/>. It is currently being hosted on a Unix server environment provided by the University of Cape Town's Computer Science Department. On this host lies an Apache web server. The Apache web server software handles incoming requests and directs traffic flow to the

'web container' Docker instance. This docker instance serves as a containerised environment for the web application and web portal, providing a lightweight and isolated environment where the website's code is executed. This 'web container' docker instance communicates with the 'db container' Docker instance that contains the PostgreSQL database. A diagram of the configuration is found in 2.

4 METHODOLOGY

4.1 Workshop Sites and Participants

Co-design is the process of collaboratively designing a product with the input and involvement of the stakeholders [6]. Ethics clearance was applied for and obtained from the University of Cape Town's ethics committee to navigate this interaction. This methodology aimed to involve the stakeholders in reiterating their software and evaluating the system's performance in diverse digital maternal and child health contexts. Throughout the project, co-design was enacted through communications and co-design sessions with the relevant stakeholders through workshops. The primary stakeholders for this project are the BBP staff and beneficiaries, with whom we engaged at the beginning and end of the project. We also involved the Sweetwaters community as additional stakeholders, whose contributions are relevant for analysing the system's performance within another context. BBP participants were chosen using convenience sampling, where available participants were asked to participate, and those from Sweetwaters, through leveraging an existing connection from exploring maternal and child health. Throughout all of the co-design sessions, data was collected through audio recordings, notes, and surveys to capture the participants' feedback, ideas, and preferences. At the beginning of every workshop, informed consent was obtained from the participants. Participants were presented with two ethics forms each, one for the researchers and one for themselves. The details of the form were explained, and participants were asked if they were comfortable with being recorded by audio, photograph or video. Participants were comfortable doing so for all workshops.

4.2 Workshops and Interviews

4.2.1 Introductory meeting (24 April) An introductory meeting was held to introduce the team and gain a better understanding of the work done by BBP. The meeting was held at the BBP offices and consisted of two BBP staff members and the team involved with the project. The BBP staff members involved were two female therapists, one of which is a speech and language pathologist and the other a physiotherapist. These staff members are the ones who engaged with us throughout the project. In this meeting, the team discussed their responsibilities in the project, and the staff briefly described their experience with the application. Thereafter, plans to conduct an interview with the staff and a workshop with the caregivers were made for the 5th and 6th of May, respectively.

4.2.2 WS1 Staff Interview (5 May) This interview centred around investigating the organisation's use of the You and Your Baby system, both from their experience using the web portal as well as their views on the application and its future. The staff interview was scheduled to be held over Zoom. Two female therapists from BBP participated.

4.2.3 WS2 Caregiver Workshop (6 May) This workshop aimed to investigate the usage and usability of the You and Your Baby application by the caregivers. This caregiver workshop was held at the BBP offices, and the participants involved were two BBP staff members and seven caregivers, four black women, one coloured woman and two black men. The first part of the workshop consisted of caregivers interacting with the mobile application, and the second half was a discussion of their experience and opinions on the application. At the end of the workshop, caregiver participants were given an honorarium of R80 each.

4.2.4 WS3 Sweetwaters Workshop (28 July - 30 July) This workshop aimed to investigate the usability and feasibility of the system in a similar context. This workshop was a 2-day workshop held in Sweetwaters in Kwazulu Natal. The participants of this workshop were black women who were mothers. Four of the participants were under the age of 35, and another four were over the age of 60. The first day of the workshop consisted of semi-structured interviews and a discussion on healthcare-seeking and how the participants navigate seeking healthcare advice. After that, cognitive walkthroughs of the mobile and web applications were done. The second workshop consisted of 7 of the 8 participants of the previous day. This workshop consisted of a cognitive walkthrough of the mobile application, updated with changes based on the last day's findings. After that, a content creation session was held, where participants gave ideas of content they would like to see on the platform and created the content themselves. At the end of each workshop, participants were given an honorarium of R170 each.

4.2.5 WS4 Final Staff Usability Workshop (10 August) The purpose of this workshop was to present the staff with the new web portal as well as to assess its usability. This usability workshop was held at the BBP offices. The participants were the two female staff members engaging with the team throughout the project. The participants were led through a cognitive walkthrough wherein they interacted with the new web portal and completed tasks to assess the new system's usability.

4.2.6 WS5 Final Caregiver Workshop (19 August) This workshop was held to obtain user feedback on the new You and Your Baby system. This caregiver workshop was held at the BBP offices. The participants of this workshop were two BBP staff members as well as five caregivers, three black women, one coloured woman and one black man. The workshop structure was a discussion, a web application cognitive walkthrough, a mobile application cognitive walkthrough and a content creation workshop. At the end of the workshop, caregiver participants were given an honorarium of R80 each.

5 DESIGN AND IMPLEMENTATION

5.1 Project Management

The project management methodology used for this project was an agile scrum methodology. The reasoning behind this was to ensure the project's iterative development and hold ourselves accountable to complete our tasks. This was important as the nature of the project divisions was highly coupled. The iterative approach allowed us to adhere to changing demands. Regarding the web portal

and web application, the dependencies to the other projects included designing the database to be used, planning how the mobile application would communicate with the web application to access resources and planning workshop logistics, such as travel arrangements, workshop protocols and workshop tools. Daily standup meetings were held over WhatsApp to update each other on each other's progress. Sprints were two weeks long each, and at the end of each sprint, a sprint review would be conducted, wherein integration would also be done whilst meeting over Microsoft Teams. Throughout our project, we made use of the Git version control system, and we hosted the code on GitHub. This allowed us to facilitate code reviews amongst each other, hold each other accountable, and keep each other up to date with each other's code base, as we would need it to understand how to navigate our dependencies.

5.2 Requirements Gathering

The WS1, WS2 co-design engagements and the 2022 System Architecture analysis revealed how the system was faulty in different areas and how this affected user engagement. They highlighted that users and staff dealt with issues accessing content on the mobile application, navigating the unintuitive and cluttered web portal, and searching through content. It also revealed that BBP beneficiaries were interested in creating their content. Additionally, a backend analysis revealed integration issues between the mobile application and the web portal, which rendered the mobile application unusable. As this is an ongoing project, the system's requirements remain the same as the previous project conducted by the 2022 Honours students [12]. The only new requirements were rectifying the identified issues and improving the system's sustainability.

5.2.1 2022 Functional Requirements The functional requirements of the web portal included the ability for BBP administrators to create and oversee both content and user management, as well as assign content to users. Meanwhile, the mobile and web applications were expected to offer user authentication, the display of both public and assigned content, and support for multi-lingual content presentation.

5.2.2 2022 Non-Functional Requirements The non-functional requirements required by the system include:

Portability and Availability: The system should be accessible from anywhere at any time.

Scalability: The system should be able to handle growing Content and User data.

Reusability: The system should be able to be reused in different contexts by different NGOs.

Interoperability: The mobile application should be able to reach the web application through an interface.

Reliability and Integrity: The system is expected to perform its intended tasks without failure or error.

Usability: The system should be intuitive and usable for the users.

Responsiveness: The system is expected to respond to the specifications of the different devices it would be used on.

5.2.3 Emerging Functional Requirements The emerging functional requirements of the system encompass several crucial aspects, including developing a more user-friendly web portal. Additionally, the requirements aim to address the challenges associated with the integration issues that have sometimes hindered the mobile

application's access to content. Moreover, the poorly structured JSON responses from the API are to be resolved, and a robust user authentication mechanism is to be implemented to enhance overall system functionality.

5.2.4 Emerging Non-Functional Requirements The emerging non-functional requirements of the system are reliability and integrity in the form of resolving the integration issues that render the mobile application unusable at times.

5.3 Development Tools

In developing the system, the choice of development tools proved to be a critical aspect of our software development life cycle. Our goal was to use the most effective development tools to ensure effective team collaboration and allow us to deliver a functional system.

To create a flexible API that allows data exchange between the system and the mobile application, the Django REST Framework (DRF) was used. The DRF offers valuable features such as automatic serialisation that allows us to return content data in simple JSON formats, which solves the issues of the dirty API returns of the previous system [2]. The DRF authentication mechanisms provide secure API access using token-based authentication [2]. The API endpoints were created with minimal code through DRF viewsets and serialiser, resulting in cleaner and more maintainable code.

Different hosting platforms were used throughout the progression of the project. In the early period of the project, a MariaDB database was hosted on Amazon Web Services (AWS). For the first usability workshop, the web application was hosted on Heroku, a platform as a service (PaaS) cloud platform [11]. After that, the web portal was added to the website, and due to a Python version issue between Heroku and the DRF, Render became the new host platform for the system, with the Postgres database being hosted on AWS. Due to integration issues between the system platform and the BBP's website server, the website and database are currently hosted on the UCT network.

The web portal and web application were built using the Django framework. Django is a Python web framework that allows for the rapid development of secure and maintainable websites [1]. Therefore, the Django Framework was the most suited based on this project's security and maintainability requirements. Another feature of the framework is its portable nature, which would allow the system to meet the need for portability and enable it to be used in different contexts.

5.3.1 Django justification The most important development tool consideration is that which the system will be built on. The decision to build the system using the Django Web Framework was based on how the capabilities of this framework could fulfil the system's predefined requirements. The following factors drove the suitability of the framework:

User Authentication: Django provides user authentication and authorisation features. This aligns with the requirement to authenticate web and mobile application users.

Content and User Management: Django's built-in admin interface and ORM (Object-Relational Mapping) layer provide an interface to make it easy to create and manage models, specifically content and user models, meeting the requirement of the system

being able to create and manage content and users for BBP Administrators. Django's architecture allows scalability through load balancing and optimising database queries to handle growing content and user data.

Portability and Availability: Django applications can be deployed on hosting platforms and cloud services, making them accessible from anywhere at any time.

Reusability: Django's modular design and separation of concerns make it possible to reuse components or the entire system in different contexts, aligning with the goal for reuse by other NGOs.

API Design: Django provides RESTful API development capabilities, facilitating mobile application and web portal interoperability. Django can structure and optimise JSON responses from the API, enhancing data exchange between the mobile application and web portal.

User-Friendly Web Portal: Django's flexibility allows for creating a more user-friendly web portal by enabling the customisation of the interface.

5.4 Implementation

Sprint One (28/04 - 12/05): At this point, an ideal software architecture was planned. The tech stack was decided as Django with a MariaDB database. This sprint was also spent learning the new framework and how to use servers.

Sprint Two (13/05 - 26/05): We had planned to create an intermediary for the mobile application to communicate with the database through an API. The MariaDB database was hosted on AWS for development purposes to allow the web and mobile applications to be developed before the API was created.

Sprint Three (27/05 - 9/06): Creation of the web application and web portal. The mobile application encountered problems with the MariaDB database, so we migrated to a Firebase database.

Sprint Four (10/06 - 23/06): The Django web portal did not integrate with the NoSQL Firebase database. Alternative approaches were being looked into.

Sprint Five (24/06 - 7/07): Bootstrap was used to develop the user interface of the web application.

Sprint Six (8/07 - 21/07): The front end of the web application was complete. The NoSQL database compatibility issue with the Django web portal was still ongoing. Still, the Firebase database was kept for the workshop in Sweetwaters from the 28th of July to the 30th of July. This is because no web portal functionality was necessary for the workshop in Sweetwaters.

Sprint Seven (22/07 - 4/08): A usability workshop was held in Sweetwaters, KwaZulu Natal, between the 28th and 30th of July to analyse the feasibility of placing the You and Your Baby application in another setting. Concerning the web application, a cognitive walkthrough was conducted to assess the usability of the web application. The web application's database engine was switched to Postgres. An API was created using the Django Rest Framework to allow the mobile application to gain database access through the web application.

Sprint Eight (5/08 - 18/08): A usability workshop for the web portal with BBP staff was held on the 10th of August. The results of the cognitive walkthrough conducted can be found in section A - Web Portal Cognitive Walkthrough of the appendix. The final application integration was meant to take place this time, but we

ran into compatibility issues with the servers used by the BBP.

Sprint Nine (19/08 - 1/09): A usability workshop for the web application and BBP beneficiaries was held on the 19th of August to assess the usability of the web application. The results of the cognitive walkthrough conducted can be found in section B - Web Application Cognitive Walkthrough of the appendix. The final paper draft was written up.

6 TESTING

Testing was conducted throughout different stages of the project. Unit testing was conducted once each project unit was completed. This was to test the functionality of the piece of code before moving on to the next stage. After that, functional testing for each function was conducted as soon as each function was completed, and a system function test was conducted to test the entire system functionality before each workshop. As there were different iterations of the system used for different workshops throughout the project, the functional tests were done multiple times to ensure that the system functioned as it should for each iteration. After that, usability testing was done with participants to test the functionality and usability further. Once the creation of the API was completed, an API test was performed to assess its performance by the mobile application.

6.1 Unit Testing

Tests were divided into three categories: Model tests, View tests and URL tests. These components found in the system apps are essential to the system's functioning; therefore, ensuring these components are working as intended was vital. For the android, main and users apps, there is a tests.py file that includes the test cases for each component, and the results of these test cases can be found in section D - Unit Test results of the appendix. The tests of the test cases included model tests to validate the behaviour of the Content model created for the system, views and templates tests to test the behaviour of the views of the system under different conditions and URL tests to validate the URL configurations of the application.

6.2 Functional Testing

Because the apps perform a specific function and do not rely on each other to achieve these functions, there was no integration testing done as there were no dependencies between the apps besides simply moving to the page of an app. Functional testing was done to test the entire system's functionality against the tasks that the users will be performing using the system. A set of test cases relating to the functions it needs to perform was created for this. The results of this can be found in section F - Functional Test Plans of the appendix. The functional testing conducted for the web application (found in Web Application Functional Test Plan) was performed on different devices to test the website's responsiveness. To test the completeness of the actions conducted on the web portal, the changes were manually simultaneously checked in the database using the pgAdmin4 web-based interface and a separate browser with the web application open. By using the web interface to connect to the database, we could see if the database changes being made in the web portal are reflected in the database. A test case

only passed if the expected changes were seen in the web interface and web application.

6.3 API Testing

API testing was conducted to test whether the endpoints of the API were working and producing the expected results. Visual Studio Code and the Rest Client extension were used to do this. This extension simplified the process of API endpoint testing by providing an interactive environment within the editor. A test.rest file was created and served as a collection for defining and executing the API test cases. Each API endpoint has a test case. The results of this testing were successful and are shown in section E - API Test Results of the appendix.

6.4 Usability and Beta Testing

Throughout the project, usability testing was conducted with the system and users. The usability tests were done in the form of cognitive walkthroughs to gather user feedback on the system and assess the system's usability. Details of these tests can be found in sec:wkshps. The results of these workshops can be found in Findings. Overall, the system's usability was found to have a satisfactory response from the users of the systems. These tests also served as beta testing for the system as the system was made available to a group of users to gather their feedback and assess its performance and usability.

7 FINDINGS

7.1 Functionality and Features

In the pursuit of creating a system that meets all the predefined requirements, a vital step in the project is to compare these requirements with the achieved outcomes. Such a comparison will allow one to see the areas where the new system has succeeded and those where it has not.

7.1.1 Functional Requirements By comparing the identified functional requirements for the web portal and application to the final system, the system has met these requirements. The system allows users to create and manage content and users as well as assign content to users. To access assigned content, users are authenticated. Public content is displayed on the website with different language options.

7.1.2 Non-Functional Requirements The non-functional requirements met by the system include:

Maintainability: Once deployed, the system will require minimal maintenance by the staff and students.

Portability and Availability: Since the system is built using the Django web framework, it is portable because it is written in Python, which runs on and is supported by various platforms. Therefore, there is no restriction on a single web server platform, and the application should be compatible with most servers for most contexts.

Scalability: The system being built with Django allows it to be scalable since Django is designed to handle heavy traffic and large amounts of data.

Reusability: Since the system is portable, it can be reusable in other contexts. The system structure of having single responsibility apps also allows different contexts only to use the specific parts or "responsibilities" of the system they would like to use.

Interoperability: The mobile application can reach the web application through the API.

Reliability and Integrity: The system can perform its intended tasks without failure or error.

Usability: The system was reported to be intuitive and usable by the users.

Responsiveness: Bootstrap, a responsive CSS framework, was used for the user interface styling. The responsive nature of this framework allows the system to be responsive.

Whilst most non-functional requirements have been met, the non-functional requirements of portability and availability have not been fully met. The specific issue is because of the integration issue between the servers deploying the BBP's website. Their website is now hosted on a Cloudways server that only supports WordPress, Magento, PHP, and other eCommerce web applications. Therefore, their current server does not have support to host a Django web application.

7.1.3 Ramifications and Mitigation Due to the integration issue previously discussed, a significant ramification of this issue is that the web application cannot be hosted on the BBP's server. A possible resolution would be that of migrating to a different server. We proposed to the BBP to switch to Render, which would cost R464,50 (\$25). This, however, would not be a feasible option, as their current server is being provided to them pro-bono by Poln. At the moment, the website is being hosted on the UCT network.

7.2 Usability and User Experience

Based on the results of the series of usability tests that have taken place throughout the project, we were able to validate whether we achieved the fundamental design goal of building a user-centric system.

The BBP staff responses from the usability tests revealed that they found the new web portal to be an improvement compared to the usability of the previous one. They reported that the new web portal was more intuitive due to the more straightforward and less cluttered interface. In addition, one staff member stated, "This system looks much better than the old one" and that it was "easier to upload". They also expressed that the system was "a lot more intuitive and user-friendly". Overall, the BBP staff reacted positively to the new web portal, affirming the success of the system's user-centric design goal. Therefore, this new web portal will enable a more efficient workflow for the staff by addressing the previous issue of the unintuitive web portal design, decreasing the time required to upload content.

Although the overall feedback from the users and the BBP staff was positive, they did have feedback on improvements and changes they would like to see. They desired for the web application to have a more colourful user interface, the ability to be able to be able to tell the format (video, image or pdf) of a content item before opening it, as well as the ability to search through content.

7.3 Sweetwaters Engagement

In the context of Sweetwaters, we decided to deploy only the web application component of the system due to the absence of a designated person who would take the content administrator role. During the Sweetwaters engagement, the web application focused

only on displaying early childhood development content from the Firebase database. The participants from this group are tight-knit and are familiar with each other. One participant stated, "Usually, we just go to different family members to get advice if the baby is sick". This group of participants rely on each other for information and guidance on navigating early childhood development. A significant finding from this interaction was that the Sweetwaters community expressed a desire to upload and download videos. They were interested in using the platform not only as content consumers but also as contributors. They were keen for the system to be a means for them to provide one another and their community with advice and content. This interest in participating in content creation validates our decision not to include a web portal in this context. Additionally, the community expressed a preference for more vibrant and colourful theme designs in both the web and mobile applications.

8 RELATED WORK

Building sustainable software for low-resource contexts is an effort that requires careful planning and execution, with a primary focus on long-term viability. To achieve this, different aspects require consideration, as outlined by Bon et al. [5]. Firstly, it requires an understanding of the digital literacy levels of the target community, as these levels play a role in influencing the software design's alignment with the community's preferences. Additionally, the availability and accessibility of technology and infrastructure the community needs to access the software resources, such as power supply and internet connectivity, need to be considered. Furthermore, the potential mismatch between the objectives of sponsors and end-users' goals could occur. Often, sponsors have broader, generic plans to increase technology resource access in developing countries, whilst end-users have specific needs for their contexts. The aim is to meet the sponsors' goals while creating valuable community services. Lastly, sustainability remains a concern in ensuring the viability of the software service being able to continue running in the event of sponsor funding being withdrawn or running out.

Building sustainable software for low-resource contexts requires a deep understanding of the specific challenges and needs of the end-users. An approach that can be taken is to co-design software with low-resource contexts. Co-design prioritizes the involvement and input of end-users and local communities in the software development process [7]. By involving the end-users who will benefit from the software, we gain invaluable insights into their unique contexts, needs, and desires, which mitigates the issue of designing a solution that aligns with the objectives of sponsors more than the end-users [7]. An instance of the use of co-design in creating software for a particular context is a study by Wardle et al. [14] that focused on the different ways in which mothers of small children interact with computers and the lack of research in designing technology for this user group. The study took a co-design approach to gain insights into these mothers' behaviours and technology usage. It specifically worked with Milk Matters, a human milk bank in Cape Town, to understand how to design for milk donor mothers. The research aimed to uncover the motivations for breast milk donation, the role of ICTs in breast milk donation, and the challenges of co-designing with breastfeeding mothers.

9 DISCUSSIONS

9.1 Reflection of Architectural Decisions

The benefits experienced by the new system resulting from the change of system architecture are present in the following areas:

Customisation, Flexibility and Modularity: Django offered better theme and functionality customisation and flexibility than WordPress. While WordPress provides themes and plugins for customisation, the use of Django allowed us to build a completely customised application tailored to our specific project requirements. This flexibility extends to future development, enabling future developers to further customise the system without being limited to available themes and plugins. Additionally, Django encourages a modular app-based design, making it easier to organise the new system, contrasting with WordPress's reliance on themes and plugins, sometimes resulting in a less organised codebase.

Security, User Authentication and Authorisation: Django provides built-in protection against common web vulnerabilities. In contrast, WordPress requires additional security plugins and measures to achieve a similar level of protection. The previous WordPress iteration of the system had no security measures in place. Moreover, Django offers comprehensive authentication and authorisation features, allowing one to manage user roles and permissions effectively. The previous WordPress iteration did not have any authentication or authorisation mechanisms. These authentication and authorisation mechanisms are present in the system in areas such as API interactions requiring authentication and users having to log in before viewing assigned content.

API Development: Django has built-in support for creating RESTful APIs, which is essential for providing data to the mobile application. While WordPress can provide APIs through plugins, Django's API development capabilities allow control, customisation, and flexibility in the API design. Given this added control, we were able to create a custom API for the system, improving the communication between the web and the mobile application. We tested the performance of the API and found it to be an improvement over the WordPress API as it was simpler and functional and provided the mobile application with readable responses.

Database Management: Django's Object-Relational Mapping (ORM) layer simplifies database management by abstracting database interactions. It provides better support for complex data structures and relationships than WordPress, which primarily relies on custom fields and plugins.

Deployment Options: Django can be deployed on various hosting environments, including traditional web servers and cloud platforms. This flexibility can allow different organisations to choose the hosting solution that best suits their needs and budget to adopt the system. However, it should be noted that one shortcoming of this project in this area is the incompatibility of the Django system with the hosting server of the BBP's current website, and therefore, it could not be deployed by the organisation.

Overall, the primary advantage of opting for Django for the new system is the aspect of control. Using a web development framework such as Django to create the system instead of WordPress, we gained the ability to have complete control over the system's features and functionalities, a level of control that was previously unattainable.

9.2 Reflection on Project Management Practices

In addressing the challenges You and Your Baby faced, the execution of effective project management practices played a vital role. The executed project management process allowed us to produce a working system that addresses the needs of the BBP. The iterative agile approach was implemented well in terms of the system, which is reflected in the final design. Considering that throughout the project, the team was spread across locations and was not readily available to meet simultaneously, the technologies utilised, such as Github, Microsoft Teams, and WhatsApp, allowed sufficient collaboration where there were dependencies. However, the area of the project management process that fell short was risk management. Due to the limited supported platforms of the server being used by the BBP, the final system could not be deployed on their servers. Due to a lack of server knowledge, there was no risk mitigation for this specific integration issue; therefore, no alternative measures were implemented to address this. In addition to poor risk management, slow development is another factor that brought about the late realisation of this problem. Much time was spent integrating the web application system with the Firebase database, which is incompatible with Django's ORM layer. In hindsight, better time management would be the best mitigation strategy to avoid this. At the start of the project, we did not have enough knowledge of servers to know of this potential problem, but had we completed the system earlier, this problem would have been addressed sooner.

9.3 Student-Led Software Development

One issue with student-led software development projects is their potential for continuity problems. A short paper by Talbot and Densmore [13] describes a co-design effort between Milk Matters, a non-profit milk bank in Cape Town, and past UCT postgraduate students. The collaboration aimed to create a donor-facing mobile application to facilitate communication between Milk Matters and donors. The paper details some of the challenges faced by the university-led initiative, such as the sustainability challenge involving limited resources, time and lack of ongoing maintenance. In our project, we faced similar challenges, including a short project timeline, limited knowledge of the previous system, and skills misalignment with project requirements. Consequently, when deciding whether to continue with the previous faulty system or to create a new system, we chose to recreate it. This impacted the sustainability of the project, similar to how the system faced issues, rendering it unusable by the BBP, this newly created system is also unusable by the BBP because of the server integration issues faced. A recommendation by Talbot and Densmore [13] is proper and secure storage of project information. Based on the experience of this project, we strongly agree with this suggestion. Regarding the 2022 iteration of the You and Your Baby system, the only documentation on the previous project was the previous Honours students' project proposal and final papers. This, unfortunately, did not provide enough background on the project, and more investigation had to be conducted to understand the system. Documentation plays a significant role in the success of the project handover. Effective project handover is essential in the iterative evolution of the project as it would ensure that students new to the project can effectively learn of the context, domain knowledge, design decisions and lessons learned. We hope

that for this iteration, there is enough documentation left behind that can allow the constructive development of the You and Your Baby system.

9.4 Use of the system in different contexts

One of the goals of this system is to design and develop it so that it can be adapted and used by different contexts and NGOs. Co-design engagements were conducted with different communities, specifically the BBP and the Sweetwaters groups, to achieve this. This collaboration serves the purpose of refining the system and validates its effectiveness and performance when used within different yet analogous environments.

Through co-design engagements with these communities, valuable insights and feedback were received from the users who are the intended end users of the system. By engaging with these communities, we could see how the system's design and functionality align with the common and differing needs, preferences, and requirements of the users of these different contexts. In addition, these engagements played a significant role in the iterative development process by enabling us to build and modify the system based on input gained from user experiences. This helped the system become more robust and user-centric, allowing seamless integration into various settings and addressing the specific demands of different NGOs.

The context of the BBP community is that they are an NGO that wants to provide childcare content to its beneficiaries. Here, beneficiaries are enrolled in early intervention programmes and use the content provided by the web application to aid them in caring for their child with developmental issues. The Sweetwaters community consists of a group of caregivers who do not have children with developmental disabilities, but they use the application to access Early Childhood Development content.

Common areas of feedback that both the communities had was their wishes for the web application to be more colourful. However, because the web application's current theme is based on the logo colours of the BBP, this change has not been effected yet. Both groups also expressed interest in contributing to the content available to support one another. The BBP participants' goal was to support each other and provide each other with their knowledge and experiences with specific disabilities and medical processes or overall personal health while navigating similar circumstances.

10 CONCLUSIONS AND FUTURE WORK

In conclusion, this paper outlined the methodology employed in this iteration of the co-design and development of the You and Your Baby system. The research followed a co-design approach involving various workshops and interviews with BBP staff and caregivers, and participants from the Sweetwaters community. Using Django and other development tools facilitated the creation of a user-friendly web portal and mobile application. The transition from WordPress to Django for developing the new system has yielded a range of significant benefits across various critical areas. However, it's important to note that while most requirements have been successfully met, challenges persist in meeting the non-functional requirements of portability and availability. The specific issue stems from integration difficulties with the BBP server. Future work for this project should entail future students making further

improvements and expansions to the system, hopefully with better documentation. These improvements should include potentially modifying the web application's user interface, displaying content formats on the content, and adding search functionality to the web application.

11 ACKNOWLEDGEMENTS

I want to thank my research partners, Rinya Singh and Tariro Baganayi, for their support and collaborative efforts during the project. I wish to express appreciation to my project supervisor, Professor Melissa Densmore, for her valuable guidance and feedback throughout the project and for the generous funding provided to enable the research to be conducted. I want to extend my appreciation to the Bhabhisana Baby Project and its beneficiaries for their involvement and collaboration on the project.

REFERENCES

- [1] [n. d.]. Django. <https://www.djangoproject.com/>
- [2] [n. d.]. Home - Django REST framework. <https://www.django-rest-framework.org/>
- [3] Rehema C. Batti. 2014. Challenges Facing Local NGOs in Resource Mobilization. *Humanities and Social Sciences* 2, 3 (2014), 57. https://www.academia.edu/38153454/Challenges_facing_local_NGOs_in_resource_mobilization_Published_paper_RM_pdf
- [4] Bipin Bhusal. 2017. Mobile first design: using Bootstrap. (2017).
- [5] Anna Bon, Hans Akkermans, and Jaap Gordijn. 2016. Developing ICT Services in a Low-Resource Development Context. *Complex Systems Informatics and Modeling Quarterly* No 9 (2016) (Dec. 2016), 84–109. <https://doi.org/10.7250/csimq.2016-9.05>
- [6] Ingrid Burkett. 2012. An introduction to co-design. *Sydney: Knode* 12 (2012).
- [7] Toshka Coleman, Sarina Till, Jaydon Farao, Londiwe Shandu, Nonkululeko Khuzwayo, Livhuwani Muthelo, Masenyani Mbombi, Mamare Bopape, Alastair Van Heerden, Tebogo Mothiba, Shane Norris, Nervo Verdezoto, and Melissa Densmore. 2023. Reconsidering Priorities for Digital Maternal and Child Health: Community-centered Perspectives from South Africa. 7 (2023).
- [8] Hélène Delisle, Janet Hatcher Roberts, Michelle Munro, Lori Jones, and Theresa W. Gyorkos. 2005. The role of NGOs in global health research for development. *Health Research Policy and Systems* 3, 1 (Feb. 2005), 3. <https://doi.org/10.1186/1478-4505-3-3>
- [9] Nina Hall, Hans Peter Schmitz, and J Michael Dedmon. 2020. Transnational Advocacy and NGOs in the Digital Era: New Forms of Networked Power. *International Studies Quarterly* 64, 1 (March 2020), 159–167. <https://doi.org/10.1093/isq/sqz052>
- [10] Birgit Penzenstadler. 2013. Towards a definition of sustainability in and for software engineering. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC '13)*. Association for Computing Machinery, New York, NY, USA, 1183–1185. <https://doi.org/10.1145/2480362.2480585>
- [11] Jeffery Solheim. 2015. Web apps in the computer science curriculum: a guide using Heroku, Java servlets, and Postgres. *Journal of Computing Sciences in Colleges* 30, 5 (May 2015), 126–133.
- [12] Jonathan Swanepoel. 2022. *You and Your Baby: Co-designing an ECD Content Delivery System for a NGO*. Technical Report. Cape Town, South Africa.
- [13] Deborah Talbot and Melissa Densmore. 2023. *Milk Matters 4.0: Challenges in deploying university-led mobile application development for small NGOs*. Technical Report. Cape Town, South Africa.
- [14] Chelsea-Joy Wardle, Mitchell Green, Christine Wanjiru Mburu, and Melissa Densmore. 2018. Exploring Co-Design with Breastfeeding Mothers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3173574.3174056>

12 APPENDIX

12.1 Github Repository

<https://github.com/beyonsay/BBP2-YAYB>

12.2 Host Configuration diagram

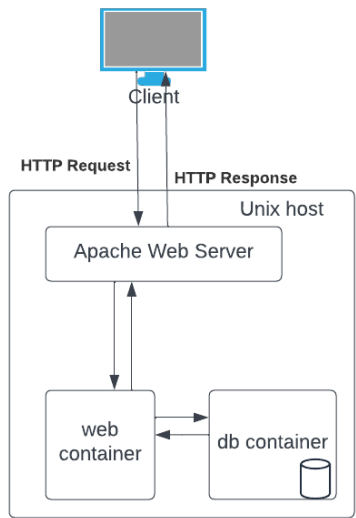


Figure 2: Architecture of current web host

12.3 Model structure

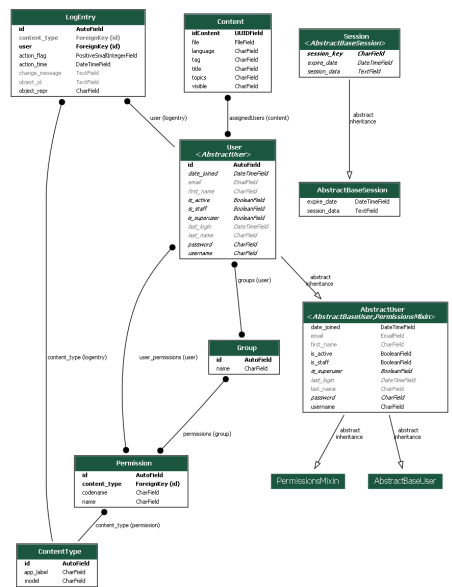


Figure 3: Model structure

12.4 A - Web Portal Cognitive Walkthrough

Table 1: Web Portal Web Application Cognitive Walkthrough

Test ID	1 - Web Portal
Purpose	Staff can log in and find the portal
Instruction	Log into the website and find the admin dashboard
Task	<ul style="list-style-type: none">(1) Click login button(2) Supply credentials and log in(3) Press the admin button
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members successfully completed the test case.

Test ID	2 - Web Portal
Purpose	Staff can navigate dashboard
Instruction	Explore the Content and Users items
Task	<ul style="list-style-type: none">(1) Click the Content and Users tabs
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members successfully completed the test case.

Test ID	3 - Web Portal
Purpose	Staff can upload new content items
Instruction	Upload a new content item
Task	<ul style="list-style-type: none">(1) Click content tab and then add content (Or add button on contents tab on the dashboard)(2) Staff to fill in required information and press save
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members successfully completed the test case.

Test ID	4 - Web Portal
Purpose	Staff can create a new user
Instruction	Create a new user
Task	<ul style="list-style-type: none">(1) Click users(2) Click add user(3) Enter username, password and password confirmation(4) Click Personal info and fill in first name, last name and email address(5) Click save
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members successfully completed the test case.

Test ID	5 - Web Portal
Purpose	Staff can assign content to a user
Instruction	Assign a (Lwazi) user to a content item
Task	<ol style="list-style-type: none"> (1) Go to contents tab (2) Click the content item (3) Click on Lwazi user (4) Click save
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members successfully completed the test case.

Test ID	6 - Web Portal
Purpose	Staff can change user password
Instruction	Update the password of the user you created
Task	<ol style="list-style-type: none"> (1) Click user tab (2) Click user (3) Click “this form” (4) Enter new password (5) Click “Change password”
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members successfully completed the test case.

Test ID	7 - Web Portal
Purpose	Staff can update content
Instruction	Change the topic for one content item
Task	<ol style="list-style-type: none"> (1) Click content tab (2) Click on a content item (3) Change the topic (4) Click “Save”
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members successfully completed the test case.

Test ID	8 - Web Portal
Purpose	Staff can delete a content item
Instruction	Delete any content item
Task	<ol style="list-style-type: none"> (1) Click content tab (2) Click on a content item (3) Click “Delete”
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members successfully completed the test case.

12.5 B - Web Application Cognitive Walkthrough

Test ID	1 - Web Application
Purpose	Users can log into the web app
Instruction	Log in to the website
Task	(1) Click login button (2) Fill in username and password and sign in
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members and users successfully completed the test case

Test ID	2 - Web Application
Purpose	Users can find a specific content category
Instruction	Find the category that will give you Baby Health content
Task	(1) Navigate to Baby Health button ALTERNATIVELY (2) Scroll to the bottom of the page and filter
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members and users successfully completed the test case

Test ID	3 - Web Application
Purpose	Users can find assigned content
Instruction	Find the content that has been assigned to you
Task	(1) Navigate to Assigned content button
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members and users successfully completed the test case

Test ID	4 - Web Application
Purpose	Users can log out
Instruction	Log out of your account
Task	(1) Click logout bottom at the top-right corner of the page
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	2/2 Staff members and users successfully completed the test case

12.6 C - Sweetwaters Web Application Cognitive Walkthrough

Test ID	1 - Sweetwaters Web Application
Purpose	Users can find a content item
Instruction	Find and play any content item
Task	(1) Navigate to a Category button (2) Select a content item ALTERNATIVELY (3) Scroll to the bottom of the page and select a content item
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	Users successfully completed the test case

Test ID	2 - Sweetwaters Web Application
Purpose	Users can find a specific content category
Instruction	Find the category that will give you Baby Health content
Task	(1) Navigate to Baby Health button ALTERNATIVELY (2) Scroll to the bottom of the page and filter
Expected Outcome	It is expected that this task is completed easily.
Actual Outcome	Users successfully completed the test case

12.7 D - Unit Test results

Test case	Result
test_assigned_view_url	Passed
test_main_view_url	Passed
test_content_view_url	Passed
test_category_view_url	Passed
test_teesandcees_view_url	Passed
test_content_view_url	Passed
test_main_url	Passed
test_assigned_url	Passed
test_android_url	Passed
test_login_url	Passed
test_logout_url	Passed
test_admin_url	Passed

Table 2: URL Test Cases

Test case	Result
test_assigned_view_with_authenticated_user	Passed
test_assigned_view_with_unauthenticated_user	Passed
test_main_view	Passed
test_content_view	Passed
test_category_view	Passed
test_teesandcees_view	Passed

Table 3: View and Template Test Cases

Test case	Result
test_content_creation	Passed
test_assigned_users	Passed

Table 4: Model Test Cases

12.8 E - API Test Results

Test case	Result
android_login	Passed
test_token	Passed
get_all_content	Passed
get_assigned_content	Passed
android_logout	Passed

Table 5: API Test Cases

12.9 F - Functional Test Plans

Table 6: Web Portal Functional Test Plan - User Data

Test ID	Test Description	Test Steps	Test Data	Expected Results	Test Results
1	Log into the web portal	Click login button, Supply credentials and log in, Press the admin button	Username: BBPAdmin, Password: youandyourbaby	Admin portal is found	Pass
2	Create a new user	Click users, Click add user, Enter username, password and password confirmation, Click Personal info and fill in first name, last name and email address, Click save	Username: LwaziSibeko, Password: youandyourbaby, First Name: Lwazi, Last Name: Sibeko, Email: sbklwa003@myuct.ac.za	Successful creation that shows up in list of users	Pass
3	Assign content to a user	Go to contents tab, Click the content item, Click on Lwazi user, Click save	Any random content item to LwaziSibeko	Content item will be assigned to user (user will show up in assigned users column)	Pass
4	Change a user's username	Click user tab, Click user, Click "Personal Info", Enter new name, Click "Save"	Change user LwaziSibeko to LwaziSibekos	Username will change	Pass
5	Change a user's name	Click user tab, Click user, Click "Personal Info", Enter new name, Click "Save"	Change user LwaziSibekos' name to Lwazii	Name will change	Pass
6	Change a user's password	Click user tab, Click user, Click "this form", Enter new password, Click "Change password"	User LwaziSibekos, new password: meandmybaby	Password updates	Pass
7	Change a user's last name	Click user tab, Click user, Click "Personal Info", Enter new last name, Click "Save"	Change user LwaziSibekos' last name to Sibekoo	Last name will change	Pass
8	Change a user's email	Click user tab, Click user, Click "Personal Info", Enter new email, Click "Save"	Change user LwaziSibekos' email address to abklwa003@myuct.ac.za	Email will change	Pass
9	Delete a user	Click user tab, Click user, Click "Delete", Click "Yes I'm sure"	User: LwaziSibekos	User removed from user table, as well as content item's assigned users	Pass

Table 7: Web Portal Functional Test Plan - Content Data

Test ID	Test Description	Test Steps	Test Data	Expected Results	Test Re-sults
1	Upload a public new content item	Click content tab and then add content (Or add button on contents tab on the dashboard), Staff to fill in required information and press save	A random pdf found on the test pc, Title: Test file, Visible: Public, Tag: Pdf, Topics: Baby Health, Language: IsiXhosa, Assigned Users: none	Successful upload that shows up in the list of content item in the web portal, web application and database	Pass
2	Upload a private new content item	Click content tab and then add content (Or add button on contents tab on the dashboard), fill in required information and select private and press save	A random pdf found on the test pc, Title: Test file2, Visible: Private, Tag: Pdf, Topics: Baby Health, Language: IsiXhosa, Assigned Users: none	Successful upload that shows up in the list of content items in web portal and database and not the web application	Pass
3	Change a content item's file	Click content tab, Click on a content item, upload a new file, Click "Save"	Test File content item being updated with a new random pdf file from the pc	Updated file in web portal, database and web application	Pass
4	Change a content item's title	Click content tab, Click on a content item, Change the title, Click "Save"	Test File content item title being updated to Test Filee	Updated title of file reflected in web portal, database and web application	Pass
5	Change a content item's visibility	Click content tab, Click on a content item, Change the visibility, Click "Save"	Test Filee content item visibility being updated to private	Updated visibility of file reflected in web portal and database and not visible on web application	Pass
6	Change a content item's topic	Click content tab, Click on a content item, Change the topic, Click "Save"	Test Filee content item topic being updated to Baby Development	Updated topic of file reflected in web portal, database and web application	Pass
7	Change a content item's language	Click content tab, Click on a content item, Change the language, Click "Save"	Test Filee content item language being updated to Afrikaans	Updated language of content item reflecting in web portal, database and web application	Pass
8	Delete a content item	Click content tab, Click on a content item, Click "Delete"	test Filee content item	Content removed from content table in database, web portal and web application	Pass

Table 8: Web Application Functional Test Plan

Test ID	Test Description	Test Steps	Test Data	Expected Results	Test Results
1a	UCT Research Project link works	Enter page link in browser and press enter	http://bbp-1.cs.uct.ac.za/	Web application home page appears	Pass
1b	Heroku website link works	Enter page link in browser and press enter	https://yayb-d953205642a2.herokuapp.com/	Web application home page appears	Pass
1c	Render website link works	Enter page link in browser and press enter	https://yayb.onrender.com/	Web application home page appears	Pass
2	Baby Development Category works	Navigate to Baby Development button	None	Leads to page with Baby Development content	Pass
3	Baby Health Category works	Navigate to Baby Health button	None	Leads to page with Baby Health content	Pass
4	Parent Health Category works	Navigate to Parent Health button	None	Leads to page with Parent Health content	Pass
5	ECD Content category filtering works	Scroll to the bottom and select filter	None	Only Baby Development content shows up under the Baby Development filter, Only Baby Health content shows up under the Baby Health filter, Only Parent Health content shows up under the Parent Health filter	Pass
6	Content items display correctly in their cells on the home and content pages	Navigate to filtering section to see how they show up and navigate to the categories to see how they display there	None	Cells contain title of content item, category, and language	Pass
7	Content page displays as should	Select a content item from the filter section of the page	None	Content page includes return button, title, topic, content media, and terms and conditions	Pass
8	Login works	Click the login button, Fill in username and password and sign in	Username: BBPUser, Password: youandyourbaby	User is logged in	Pass
9	Assigned content button shows up after logging in	Navigate to Assigned content button	None	Assigned Content button appears since user is logged in	Pass
10	Assigned content category leads to page filled with assigned content	Click Assigned content button	None	Content assigned to BBPUser shows up	Pass
11	Logout works	Click logout button at the top right of the page	None	User is logged out and logout page appears	Pass
12	Website is responsive to different devices	Open website on different devices	Devices: Windows laptop Google Chrome and Microsoft Edge, iPhone 14 Safari, Samsung Galaxy S20 Google Chrome	Website adapts to the device being used	Pass