# Capstone Project Document

Suyash Chordia: 101336884

George Brown College

Submitted To:

Dave Mckay

Abstract

Verifying a graduate student's educational credentials entails evaluating and confirming the validity of the certificate. As university administrators spend millions of dollars annually sustaining the entire process, it is an expensive, drawn-out, and time-consuming process. Additionally, the company spends a lot of time examining the applicant's certificate to ensure its validity. Traditional certificates are given to candidates by the present certification system. Certificates can therefore be altered or lost at any time. Furthermore, the process is risky due to scammers' use of fake certifications that are given by numerous illicit organizations. By forging certificates, people commonly fabricate credentials and degrees. A fake certificate generated by skillful scammers is always tough to identify and address as the original one. Therefore, there is a crucial need to upgrade the certification and verification process. This paper introduced a Blockchain-based decentralized certificates platform that offers an easy way to issue, check, and verify educational certificates.

*Requirements*

**DAPP Requirements**

Tools and Technology used in DApp:

- Blockchain Framework: Hyperledger Aries
- Language for implementing smart contracts: Solidity
- IDE for deploying smart contracts: Remix IDE
- Blockchain Network: Ropsten Test Network
- Front-end: React.JS
- Web Technology: Web 3.0
- Web 3.0 Module: Web3.JS library

**React App**

We need some dependencies to create react application. First dependency we need NPM. Node Package Manager shortly called NPM which comes with Node.js. Inject web3.js on DApp. Then after installing node.js, to create and run a fresh react app, we run this command using git or terminal in Linux or cmd in windows.

- NPM init react-app my-application, or npx create-reactapp my-application. We should use always npx. Because it always creates updated version of react.
- cd my-application
- NPM start – to deploy the application

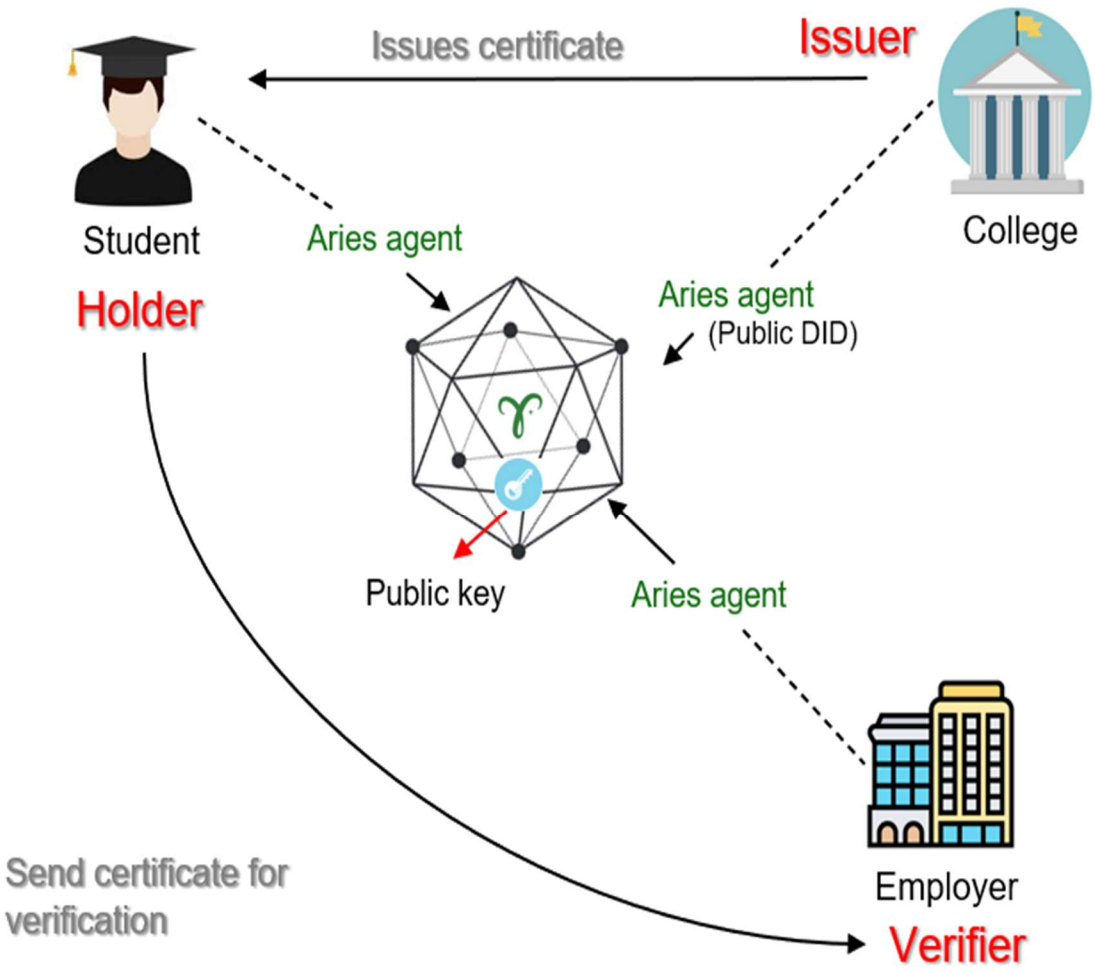
**Inject web3.js on DAPP.**

- NPM install web3

***Front end user interface***

We can create a front-end user interface for the application where the issuer, recipient, and verifier can easily interact with the Blockchain by using html, CSS, bootstrap, react JavaScript library to build the frontend user interface.

*Architecture*

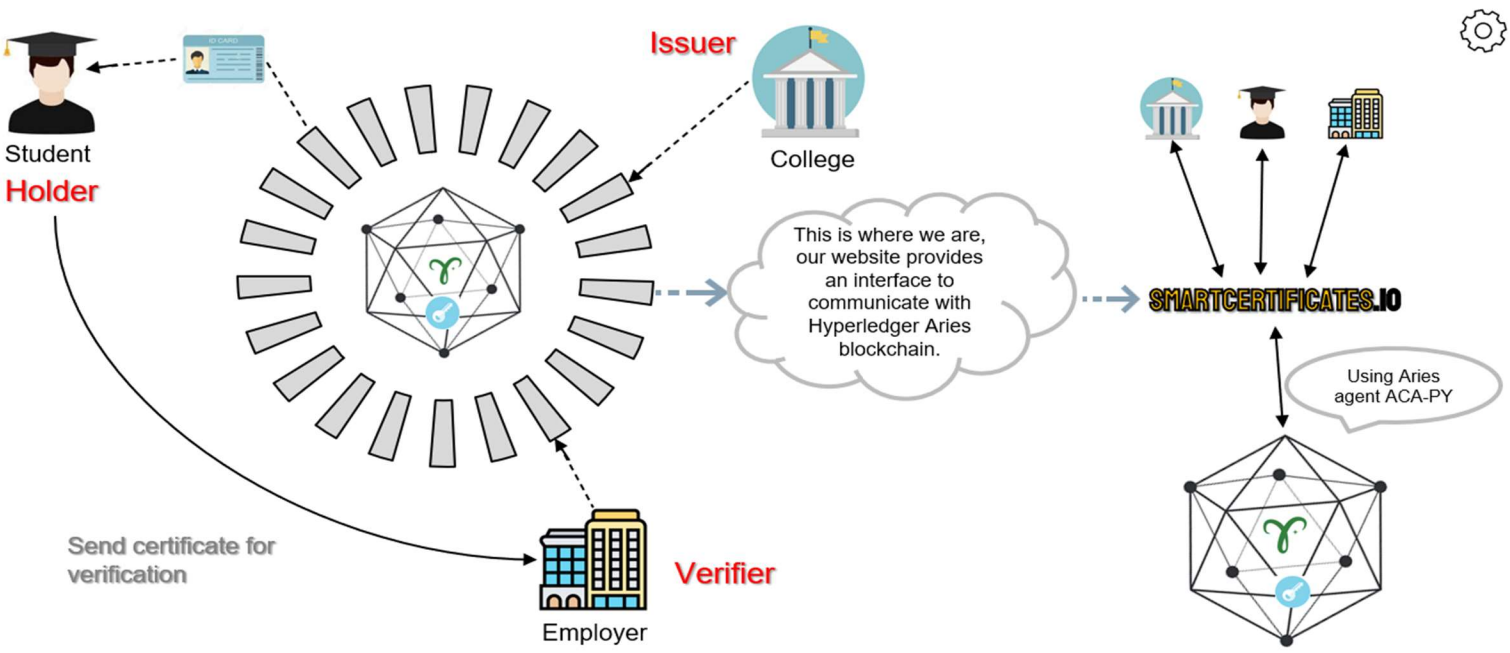



This is how current system will look like if college try to implement it, Issuer (College)first generates the certificates that contains the claims; such as: student name, course name, batch, year of passing, etc. Then they register the public DID (decentralized identifier) on Hyperledger Aries network using and Aries agent for example ACA-PY, which contains the public key and to sign the certificate they'll take hash of the claims and their private keys.

Now, holder (Student), will receive the certificate, and when they try to apply for a job, they'll share the certificate with employer.

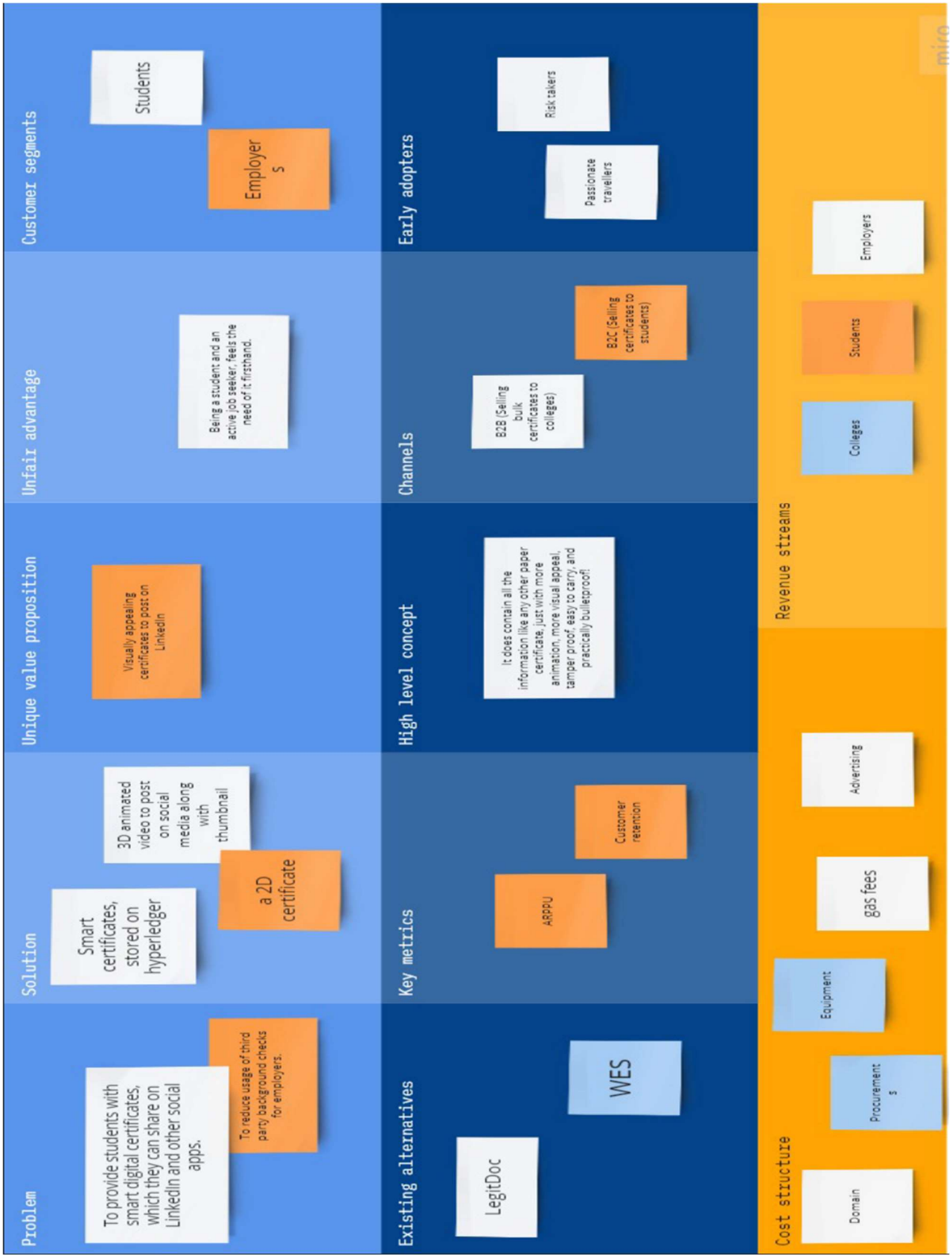
At last, verifier (employer), will request the public DID from college, and using that public DID they'll fetch the public key using their Aries agent. Now they'll have to decrypt the signature to get the hash. Now they'll match this hash with the document hash provided by student and see if it matches or not.

To reduce all these efforts from individual parties we came into the picture. Our solution is to handle the blockchain part of the process, so, all individual parties will not have to create a structure and invest their time and money into it. We'll register the public DID on behalf of the college, we'll issue the certificate to student and at last we'll verify the certificate for the employer.




So, with this structure, Issuer, Holder and Verifier, all will have to contact with smartcertificates.io and we'll in turn interact with the blockchain. Also, we'll email the certificates directly to student and not the college to reduce the number of participants the process and increase security.

*Lean Canvas*

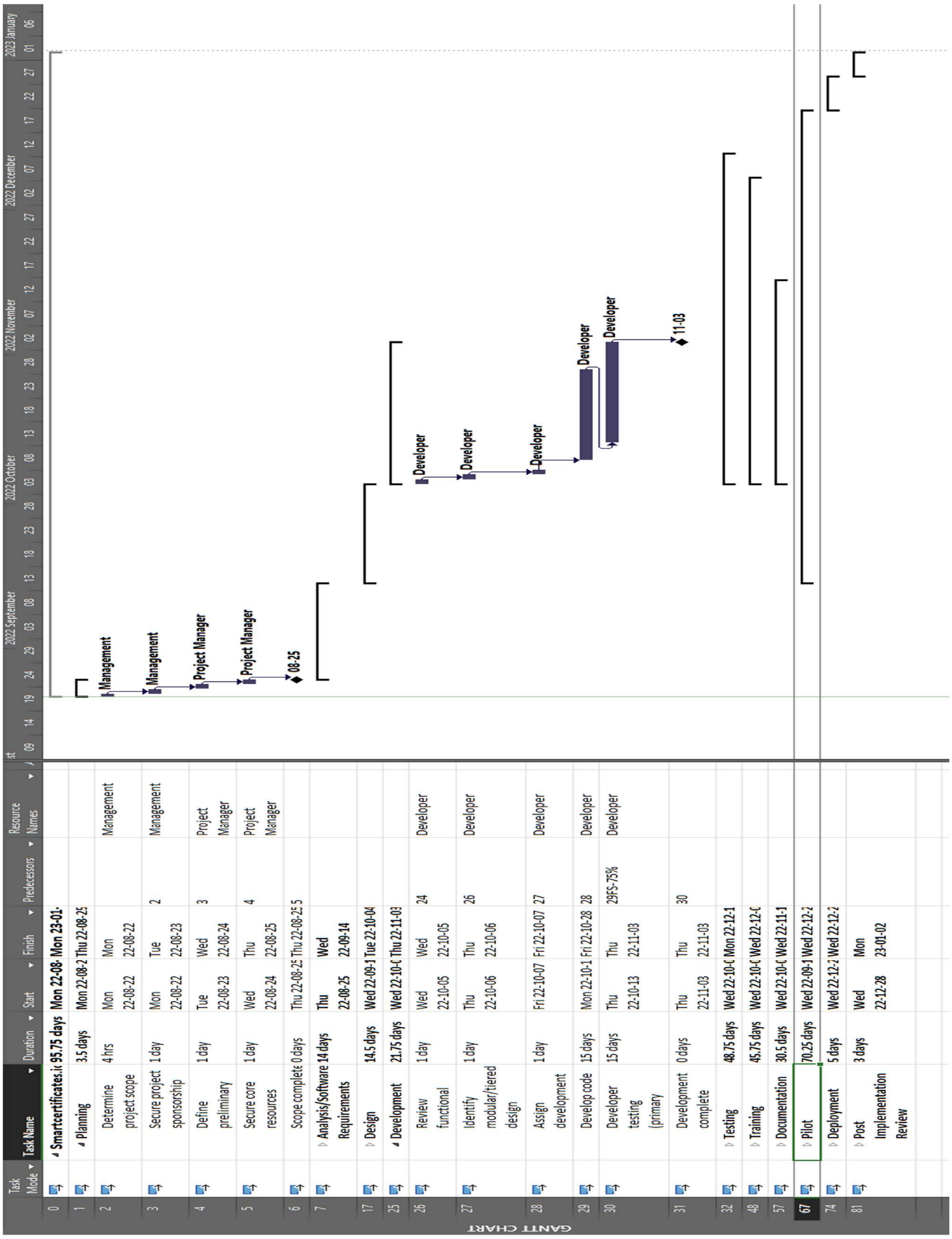

Problem
To provide students with smart digital certificates, which they can share on LinkedIn and other social apps.
Solution
Smart certificates, stored on hyperledger
3D animated video to post on social media along with thumbnail
a 2D certificate
To reduce usage of third party background checks for employers.
Unique value proposition
Visually appealing certificates to post on LinkedIn
Unfair advantage
Being a student and an active job seeker, feels the need of it firsthand.
Customer segments
Students
Employer s
Existing alternatives
LegitDoc
WES
Key metrics
ARPPU
Customer retention
High level concept
It does contain all the information like any other paper certificate, just with more animation, more visual appeal, tamper proof, easy to carry, and practically bulletproof!
Channels
B2B (Selling bulk certificates to colleges)
B2C (Selling certificates to students)
Early adopters
Risk takers
Passionate travellers
Cost structure
Domain
Procurements
Equipment
gas fees
Advertising
Revenue streams
Colleges
Students
Employers
miro

*Work Breakdown Structure*

| WBS | Task Name |
|---|---|
| **0** | **Smartcertificates.io** |
| **01** | **Planning** |
| 01.1 | Determine project scope |
| 01.2 | Secure project sponsorship |
| 01.3 | Define preliminary resources |
| 01.4 | Secure core resources |
| 01.5 | Scope complete |
| **02** | **Analysis/ Requirements** |
| 02.1 | Conduct needs analysis |
| 02.2 | Draft preliminary specifications |
| 02.3 | Develop preliminary budget |
| 02.4 | Review specifications/budget with team |
| 02.5 | Incorporate feedback on specifications |
| 02.6 | Develop delivery timeline |
| 02.7 | Obtain approvals to proceed (concept, timeline, budget) |
| 02.8 | Secure required resources |
| 02.9 | Analysis complete |
| **03** | **Design** |
| 03.1 | Review preliminary specifications |
| 03.2 | Develop functional specifications |
| 03.3 | Develop prototype based on functional specifications |
| 03.4 | Review functional specifications |
| 03.5 | Incorporate feedback into functional specifications |
| 03.6 | Obtain approval to proceed |
| 03.7 | Design complete |
| **04** | **Development** |
| 04.1 | Review functional specifications |
| 04.2 | Identify modular/tiered design parameters |
| 04.3 | Assign development staff |
| 04.4 | Develop code |
| 04.5 | Developer testing (primary debugging) |
| 04.6 | Development complete |
| **05** | **Testing** |
| 05.1 | Develop unit test plans using product specifications |
| 05.2 | Develop integration test plans using product specifications |
| **05.3** | **Unit Testing** |
| 05.3.1 | Review modular code |
| 05.3.2 | Test component modules to product specifications |
| 05.3.3 | Identify anomalies to product specifications |
| 05.3.4 | Modify code |
| 05.3.5 | Re-test modified code |
| 05.3.6 | Unit testing complete |
| **05.4** | **Integration Testing** |
| 05.4.1 | Test module integration |
| 05.4.2 | Identify anomalies to specifications |

| | |
|---|---|
| 05.4.3 | Modify code |
| 05.4.4 | Re-test modified code |
| 05.4.5 | Integration testing complete |
| **06** | **Training** |
| 06.1 | Develop training specifications for end users |
| 06.2 | Develop training specifications for helpdesk support staff |
| 06.3 | Identify training delivery methodology (computer-based training, classroom, etc.) |
| 06.4 | Develop training materials |
| 06.5 | Conduct training usability study |
| 06.6 | Finalize training materials |
| 06.7 | Develop training delivery mechanism |
| 06.8 | Training materials complete |
| **07** | **Documentation** |
| 07.1 | Develop Help specification |
| 07.2 | Develop Help system |
| 07.3 | Review Help documentation |
| 07.4 | Incorporate Help documentation feedback |
| 07.5 | Develop user manuals specifications |
| 07.6 | Develop user manuals |
| 07.7 | Review all user documentation |
| 07.8 | Incorporate user documentation feedback |
| 07.9 | Documentation complete |
| **08** | **Pilot** |
| 08.1 | Identify test group |
| 08.2 | Develop software delivery mechanism |
| 08.3 | Install/deploy software |
| 08.4 | Obtain user feedback |
| 08.5 | Evaluate testing information |
| 08.6 | Pilot complete |
| **09** | **Deployment** |
| 09.1 | Determine final deployment strategy |
| 09.2 | Develop deployment methodology |
| 09.3 | Secure deployment resources |
| 09.4 | Train support staff |
| 09.5 | Deploy |
| 09.6 | Deployment complete |
| **010** | **Post Implementation Review** |
| 010.1 | Document lessons learned |
| 010.2 | Distribute to team members |
| 010.3 | Create maintenance team |
| 010.4 | Post implementation review complete |

*Project Timeline (using Gantt chart)*



References


- https://www.georgebrown.ca/impact-report-2020-2021 drawn people starting a business project
- file:///C:/Users/16475/Downloads/12th_ICCCNT_2021_paper_192.
- https://www.youtube.com/watch
- https://www.youtube.com/watch