**Chapter 3: Stages of Development of the Smart CGPA Advisor**

**Introduction**

The Smart CGPA Advisor for Caleb University, is developed systematically in accordance to the process of Software Development Life Cycle, promoting accuracy and efficiency so that we can incorporate university grading policies and align with data structure. This chapter describes the various stages set out for the development project and is described in a systematic way based on software development life cycle (SDLC).

The primary data sources were Caleb university approved documents and databases, and not surveys or vendor analysis, since it was important to incorporate data requirements from approved information source for the the target users of the application. The process of developing the advisor has been exhilarating, yet stressful. This development project is not about coding only, but it is about knowledge of student, and faculty, needs and providing a useful tool to assist them. This has allowed the planning, creation, testing and launching of the application, to occur under SDLC, but with reduced risk and resource use. This application differs from other GPA calculators by looking other the organisations official documents and other actual databases. Every calculation is based on the university's real grading practices and academic standards (the Advisor is not dividing numbers and giving students mythical results!) and therefore accurate and meaningful, as we designed the system for a real audience.

During the stages of development (from concept to production), we will examine how the various phases of development built a tool for student, instructor and academic administrator use. In this chapter, I will take you through the well-considered reasoning and decisions made in developing the Application that established the importance of each phase.

**3.1 Stage 1: Requirements Analysis Using Documents and Database from Caleb University.**

The first stage of development is the most important stage as this is where the whole project emerges. This phase of development is an analytical overview of the existing systems and literature and our understanding of the important concepts, requirements and expected use of the Smart CGPA Advisor.

The first stage is trying to understand what you actually need to build. I began by exploring Caleb University's grading policies, student pathways and existing databases related to academic results. The main purpose of this stage was to understand the academic rules and data components associated with analysis that the Advisor should use. We comprehensively reviewed all the authoritative documents e.g., the academic handbook and grading policy manual to take everything into account to ensure that the Advisor replicated the existing patterns user needs.

I also investigated the historic processing of results. I looked for traditional ways in which errors and inefficiencies (with manual calculations) had been made, so that we could attempt to eliminate these with a system solution. For example it became apparent that reported GPAs, as a result of manual calculations often differed to flagged GPA and could have serious implications on students academic standing.

To get a more tangible sense of the user, I developed user personas form demographic data. It was important that the interface design is intuitively usable to the target users. The represents a wide range of the intended audience through to first-year students through to academic advisors. In the end, I documented everything I found and put it into a Software Requirements Specification (SRS) document that provided a touchstone for the project.

**Objective**: Define the functional and non-functional requirements of the system based on Caleb University's official grading policies, processing workflows and structure of the university's student result database.

**Activities:**

* Literature Review: Investigate different academic record management systems and associated technologies, to look for similar problems to that being solved and the mistakes that could inform a best practice approach.
* Analyse Existing Systems: Review existing ways, tools and mechanisms for GPA/CGPA calculation to determine areas of improvement.
* Functional Requirements: Determine what features where required to enable: calculations of GPA/CGPA, personalised suggestions and secure handling of data.
* Understood the Caleb University grading system, assessment processes for the student, and procedures for processing results according to the handbooks and administrative documents.
* Examined the fields and structure in the student's results database, such as biodata of students, course codes, credit units, grades, and most importantly, grade points.
* Discovered the issues with the manual processing of results such as potential miscalculated, persistent delays, and opacity.
* Identified and written requirements for the GPA/CGPA, automated calculations which will involve the existing student's database, and UI requirements for images, dimension and controls on the UI.

**Outcome** : It was important I carry out this task well as the assignment was heavily reliant on one phase leading to the next, and poor performance at any stage would imply I would have to rewrite the document later, which can be costly.

**3.2 Stage 2: System Design Based on Institutional Data**

Now having got a good grasp of the requirements, it was time for planning and design. This stage required the specification to become a feasible and useful application. I developed the system architecture design to ensure that student end-users are ultimately able to access student records in real time, and capture the necessary data ensuring the required safeguards were met.

I focused on the design of the user interface (UI) and user experience (UX).I set out to create an application that was both usable and useable. To do this, I built simple interface mockups to align the layout for data entry of student IDs, course codes, grades and credit units, and simply displayed the calculated results. The mockups were reflective of Caleb University's data structure and user flows.

I also prepared functional proofs of concept - working models of the important interface components - to show how each user would interact with the system, and to enable some initial testing and feedback which was helpful.

A couple of important design features included responsive layouts usable on any device, and accessibility features which would help ensure the application could be used efficiently by the intended audience, including users with disabilities.

Over and above the UI/UX features I also created a full database schema again tailored to efficient and effective data management between students, courses, and grades, including how relationships (and data handling) would take place in an efficient and secure manor.

**Objective**: To design the architecture and interface for the Application based on Caleb University`s data structure and grading rules.

**Activities:**

* To develop a system architecture to apply the web application to the Caleb University database for real-time data retrieval and results processing.
* Developing the user interface that defines the input fields (student ID, course codes, grades, credit units on y) and output (GPA, CGPA, personalized recommendations etc.)
* Developing database schemas and flow-charts that were reflective of the university's record-keeping processes.
* Making sure that the design incorporates methods for security and privacy of data that pertain to the use of student records.

**Outcome** : Detailed design documents, user interface mockup designs, and database schemas that are aligned with Caleb University's record-keeping policies.

**3.3 Stage 3: The implementation with Caleb University Data**

Having completed the design stage, I was on to the implementation stage of the Smart CGPA Advisor. In the implementation stage I would turn my detailed plans into a working application. I was excited and from the knowledge I had gathered I allowed the advantages of efficiency and accuracy for myself in managing academic records motivate me to design a system which would automate GPA and CGPA calculations as much as possible based on the University grading policy of Caleb University.

I developed the frontend using HTML5, CSS3 and JavaScript using Bootstrap for responsive design. The development of the frontend aligns with my understanding of user-centered design as human users come first so that accessibility and easier use were catered for, whether by students that were highly attuned to technology or not technology users altogether.I wanted to design a user-friendly interface for everyone from students who are quite comfortable interacting with technology, to those who tend to avoid technology altogether. In the background, I created the logic for the application to provide the correct calculations for GPAs and CGPAs. I also coded algorithms to understand some scenarios, such as course retakes and incomplete grades, during the requirements analysis. It was met important to be integrated with Caleb University’s current database so the application can get information from the database in real-time to avoid duplication of effort and reduce data errors.

By the time an entire exercise of building a fully functional proof of concept for the Smart CGPA Advisor had matured, it automated the processing of GPAs and CGPAs for staff and students; indeed, users had become far more comfortable with a user interface that had been simplified and made more user-friendly. This was a very important and significant milestone towards achieving an application aimed at improving academic performance management. Making a well-defined user interface is what I going to target for all users, from the technology-savvy students to those who do not know much about technology.

I have implemented courses that reflect almost all possible scenarios in courses such as retakes and incomplete grades. On the backend, however, I executed logic that calculated everything from GPA and CGPA the nearest to 100 percent accuracy. Integration with Caleb University's existing databases also made data retrieval easier and real-time, cutting out the manual errors.

I was focused on writing good quality code during the implementation phase. Keeping up with the high standards of documentation according to industry standards and best practices for developing software. Version control systems are employed in the course to organize the work, so that changes can be documented later for full accountability. Quality can also, say definitionarily, be from iterated development that leads to an application that can be improved through continuous processes of user feedback and modification.

I kept up a high standard both of code and documentation throughout my implementation. I applied best practices and used version control, such as Git, which organizes my development and makes future revisions manageable. I always pushed myself to improve, reflecting the belief I have in the iterative process of developing, which talks about enhancing an application through user feedback.

By the time an entire exercise of building a fully functional proof of concept for the Smart CGPA Advisor had matured, it automated the processing of GPAs and CGPAs for staff and students; indeed, users had become far more comfortable with a user interface that had been simplified and made more user-friendly. This was a very important and significant milestone towards achieving an application aimed at improving academic performance management. Making a well-defined user interface is what I going to target for all users, from the technology-savvy students to those who do not know much about technology.

I have implemented courses that reflect almost all possible scenarios in courses such as retakes and incomplete grades. On the backend, however, I executed logic that calculated everything from GPA and CGPA the nearest to 100 percent accuracy. Integration with Caleb University's existing databases also made data retrieval easier and real-time, cutting out the manual errors.

I was focused on writing good quality code during the implementation phase. Keeping up with the high standards of documentation according to industry standards and best practices for developing software. Version control systems are employed in the course to organize the work, so that changes can be documented later for full accountability. Quality can also, say definitionarily, be from iterated development that leads to an application that can be improved through continuous processes of user feedback and modification.

I kept up a high standard both of code and documentation throughout my implementation. I applied best practices and used version control, such as Git, which organizes my development and makes future revisions manageable. I always pushed myself to improve, reflecting the belief I have in the iterative process of developing, which talks about enhancing an application through user feedback.

I began the testing phase with unit testing of the individual components of the application to verify that every function was able to do what it was supposed to do, which included the GPA calculations and some functions that retrieved data from the database. Next, I conducted the integration testing to verify the application was appropriately communicating from the front-end to the back end. The integration testing confirmed that when the user interacted with the interface the appropriate data was sent to the database and the user interface provided the expected data at the user level. I was pleased to see that the application met the requirement specified in the SRS document generated in the requirements analysis.

For the user acceptance testing (UAT) phase, I acquired a set of user participants that consisted of the end-users, these participants included students and administrative staff. The user data and information from their navigation of the application made a world of difference in discovering usability issues, as well as, confirming that the functionality being incorporated fulfilled the real-world needs of users. Observing the user navigate and interact with the application interface allowed me to also enhance the application use of interface design and to improve usability in general.

Performance testing was conducted to evaluate the application's responsiveness with varying loads. Specifically, to simulate multiple users performing the same action at the same time to see how the application responded to multiple requests. This testing was important to determine if the application was capable of supporting such a large student body at Caleb University.

The testing phase led to a complete report covering issues, severity of issues, and accepted fixes. The report showed that all critical bugs were fixed, and the application was improved from user feedback and was ready to deploy.

**Objective** : The goal of the system testing were to ensure system correctness and reliability and usability using actual and/or sample data from the Caleb University database.

**Activities**:

* Functional testing using actual student records, to confirm that GPAs/CGPAs are being calculated correctly.
* Usability testing was conducted using academic staff and students to ensure that the interface was usable, and functioning correctly for users.
* Data security and privacy testing to verify that security and privacy comply with institutional policies.
* Verify that there were no discrepancies or bugs in calculating logic or data from the database.

**Outcome** : A secure, reliable, functional, and error-free application is suitable for Caleb University's environment and ready for use.

**3.4 Stage 4: Further Smart CGPA Advisor Functions Feature CGPA Improvement Methods and Tips**

The Smart CGPA Advisor underwent radical change in Stage 4 as it transitioned from a Smart Gadgets tool designed for grade calculation to a student’s scholarly aid. This stage worked on carving capabilities of the Advisor so students’ performance can be improved through appropriate guidance. Based on a student’s GPA and CGPA figures, the Advisor was able to tell where there was an improvement in their performance, and where they were bound to face difficulties. It was never about figures; instead, it unlocked a new vista of appreciating strengths and working towards developmental needs.

Smart CGPA Advisor allows setting of academic goals, and by guiding towards proper planning of activities necessary in achieving such targets .It also offers specialized area of concentration and supporting targeted suggestions along with tracking results. This way, the Advisor constantly refocused his strategies to better support the academic needs in which students actively sought out guidance throughout their educational pursuits.

In this stage, my focus was on giving Smart CGPA Advisor more functionality that prepared him to automatically give lecture advice alongside exercise execution on core capability improvement areas.

**Activities**:

* I spent a lot of time building algorithms that could actually make sense of how students were performing—things like grades, progress, and patterns in their academic behavior. The idea was to move beyond generic advice and instead offer suggestions that felt relevant and useful to each person, not just based on numbers, but on what those numbers meant for them individually.
* I also made the interface. Kept it pretty simple—just wanted something students could actually use without getting confused. You can see your progress, what you’re doing right, and where you need to improve.
* Talked a bunch with the academic staff to make sure whatever the system suggested wouldn’t clash with school rules. Also tried to make use of the support stuff the university already has.
* We added a place where students could leave feedback. That really helped, since people don’t always use tools the way you expect. A lot of changes came from what they told us.

**Outcome**: By the end of Stage 4, the Smart CGPA Advisor had grown into a reliable academic companion. It offered students clear, personalized advice that aligned with their individual academic paths. This transformation made the app genuinely useful—students felt more supported, more involved in their progress, and better equipped to make informed decisions about their studies at Caleb University.

**3.5 Stage 5: Launch and Feedback with Caleb University**

The launch phase was closely preceded by the deployment phase where the Smart CGPA advisor moved from development to production. We planned this phase, carefully, so that we created opportunity for a smooth roll out, and supported users in orienting themselves to the new tool.

I started with a pilot testing phase identifying a small number of users to pilot the app and interrogate it in a production environment. The pilot period eased the way to find any residual errors in the tool, and permitted me to get pilot users feedback simultaneously, which benefitted circulating the application further.

Once, ego a successful pilot testing by the willing users, we deployed the application for all other students and staff who had access to the play store for their own beneficial reasons. I made sure to have communication established clearly before hand and establish a safe way to introduce new users regard access, and tool features, and historical details on how to develop academic success as a student. I was able to overlap my introductions at cross banners-email exchanges, and weekly updates I provided these details at academic meetings, and I made posters for faculty to use as quick references around campus.

User manuals and online documents are provided to offer continuing support. Manuals and documents are designed to be as clear and accessible as possible to ensure that users of all technical backgrounds could use the system.

Systems for feedback were established to capture user feedback in a continuous manner. Surveys and contact personnel made it possible for students and staff to report issues and suggest enhancements to the system in an open collaborative manner.

User experiences were positive during the first launch within Caleb University. Students praised the platform for its usefulness with accurate GPA and CGPA calculations, and academic advisors liked that the system made customized academic proposing of advice possible.

**Objective** : The aim was to release the Smart CGPA Advisor to Caleb University students and staff while also improving the system in response to institutional feedback.

**Activities**:

* Starting with a pilot release as a preventative measure before the full rollout.
* Deploying the application on the university web systems.
* Migrating the application from development to the live production servers.
* Publishing training materials and documentation for users.
* Making the necessary equipment and resource materials available to end-users and administrators to ensure meaningful use of the application.
* Collecting feedback from users (students, academic staff, administrators) as there would be slight adjustments needed.
* Rolling out continuing updates and improvements based on naturally occurring experiences and input.

**Outcome** : A Smart CGPA Advisor that has gone live and is continuously improved and updated that is embedded into the academic ecosystem of Caleb University.

**Overview: Steps through Development**

Development of the Smart CGPA Advisor for Caleb University will take place in a linear fashion using five steps that are supported by the official documentation and database of the University. The steps will guarantee the system is responsible, usable and feasible from an institutional policy perspective. This also guarantees the solution to automating the GPA and CGPA calculation process.

|  |  |  |
| --- | --- | --- |
| Step | Tasks/Product | Outcome/Product |
| Requirements Analysis | Literature review, existing system review , SRS document | Software Requirements Specification (SRS) |
| Planning and Design | Project planning, architecture, UI/UX design , prototyping | Planning and Design Documents |
| Development | Coding, module integration, documentation | Functional application modules |
| Testing & QA Unit, integration, system and user testing | application that is tested and bug free | Deployment |
| Production setup | application is live |  |

**Conclusion**

The development of the Smart CGPA Advisor is a structured and organized plan that greatly emphasizes planning and analysis. The ability to plan and analyze the process model within the Software Development Life Cycle, from requirements elicitation to maintenance allows for a systematic approach that allows for the reduction of risks and also the continual achievement of user requirements delivering a product of value!

Not only will the Smart CGPA Advisor automate the convoluted process of GPA and CGPA calculation, it will analyze student grades to provide actionable advice to support the improvement of student academic performance. Furthermore, the Smart CGPA Advisor has a beautiful user interface and can perform a range of functionality. Additionally, the Smart CGPA Advisor can be tailored to the requirements of the educational institution, marking a critical juncture in enhancing management of an institution's academic records and way of engaging students.

Having active maintenance and updates being a recurring process based on user feedback, the Smart CGPA Advisor can be relied upon to give evidence-based suggestions to academic institutions based on a given number of academic students as their academic metrics contributions to a field they specialize in will go through constant changes in the surrounding educational environment.

Overall, there is nothing more than the thoughtful amalgamation of software engineering approach, educational philosophy, and user experience with your user experience leading to great outcomes in the sphere of educational technology, which is demonstrated through the process of developing the Smart CGPA Advisor presented in Chapter 3 in conjunction with the basis of Chapters 1 and 2.

The process of developing the Smart CGPA Advisor from a documentation perspective outlined in Chapter 3 was a thoughtful amalgamation of the software engineering approach informed by educational philosophies. In taking the initial step to understand the diversity of grading systems within institutions that students are educated in (Chapter 1) to understanding artificial intelligence fostered engagement to improve academic performance (Chapter 2), this project used technology as a means to resolve to problems faced by academic institutions concerning the management of academic records.

By combining automated GPA and CGPA calculations with tailored, actionable recommendations, the Smart CGPA Advisor allows students to take control of their journeys in higher education. The advisor structure and its underlying backend reflect the usability and fidelity to an academic context -- as recommended in Chapter 1.

Moreover, the combination of AI-driven personalized feedback and analytics with predictive analytics supports the contemporary student-centric approaches to education. As discussed in Chapter 2, contemporary approaches to improving student motivation and effort support a model of personalized, self-regulated, and proactive academic initiatives. In addition to supporting institutional efficiency and effectiveness, the system also supports transparency and accountability within the institution.

At the end of the day, the Smart CGPA Advisor illustrates possibilities for innovations to effectively improve academic performance, when the co-design of the technology is intentional, appropriate to the institution, and draws on evidence from academic learning research. The Smart CGPA will provide additional potential for later innovations to better bridge the gap within post-secondary education between data, AI, and human-centered support for student learning.