

Design and Implementation of a Software Application for Progressive Collapse Analysis of Ship Hull Girder Structures

Ikue-John, Hope^{*1,2,3}, Antai, Emmanuel¹, Okoro, Hope³, Udo, Anietie Effiong^{1,2,3}

¹Department of Marine Engineering/ Naval Architecture, Faculty of Engineering, Akwa Ibom State University, Ikot Akpaden, Akwa Ibom State, Nigeria

²Department of Marine Engineering, Faculty of Engineering, Rivers State University, Port Harcourt, Nigeria

³Department of Marine Engineering, Faculty of Engineering, Niger Delta University, Wilberforce Island, Bayelsa State, Nigeria

Corresponding Author: Ikue-John, Hope

ABSTRACT

This paper deals with the development of a software for progressive collapse analysis of ship hull girder structures. The progressive collapse phenomenon of the ship structures is a design problem that requires proper analysis to ensure structural integrity, reliability and safety of the ship throughout her lifecycle. The ship structure may deteriorate in service due to several factors. An aging ship is often exposed to corrosion as its operational environment even increases the structural corrosion. Fatigue crack and corrosion-enhanced fatigue becomes eminent as the ship continues in operation. These factors have called for concern in the design cycle to develop tools that could assess the health of the ship structures at the preliminary stage of the design. To address, this issue, several methodologies were used in the development and implementation of the software application. The program was designed in a general-purpose programming language that is platform independent. In this case, Java programming language was used which offers so many benefits well known in the information system. Various algorithms such as Agile method and evolutionary process flow were implemented in the developed program. Both hand typed inputs and imported file inputs were used to supply the required variables while the users enjoy the graphical interface of the program. The outputs of the program demonstrate the longitudinal strength of the hull girder by computation of the bending moments at specified station and the stress/strain of the primary structural elements (beams and columns) is also monitored in different failure modes at a given time. The developed program is currently running on the testing phase to ascertain its reliability and robustness with field measurement.

KEYWORDS: Design and Implementation, Progressive Collapse Analysis (PCA), Ship Hull Girder, Structures, Ship Design

Date of Submission: 05-07-2021

Date of Acceptance: 18-07-2021

I. INTRODUCTION

Over the years, the use of computer software has been pervasive in all areas of human endeavor especially in the field of engineering. Its utilization in engineering has led to the development of applications that have helped in the design, analysis, and simulation of engineering models. This has brought about an increase in productivity and efficiency in the work products of engineers. This report focuses on the design and development of a computer software or system for the analysis of the progressive collapse of ship hull girder structures.

The ship hull girder structure consists of different structural members (plating and stiffener) of different geometries and material properties welded together [1]. These structural members provide the strength or capacity for vessels (or ships) to withstand different loading conditions. Hence, it is critical to determine the response of the structure under different loading conditions and the maximum loading condition that the hull girder structure can bear. The absence of such analysis may result to failure of the hull girder structural elements as a result of loading the vessel beyond its maximum load-carry capacity, which may ultimately lead to the collapse of the vessel.

Hughes reported a fatal accident of a ship that broke down as result of its hull splitting into two and this led to total loss of the ship [2]. The report further showed that the accident was caused by excessive loading of the vessel which subjected the vessel to a bending moment that surpassed its ultimate strength. To avert the occurrence of such accidents and the attendant loss of lives and properties that come with, a proper design and

analysis of the hull girder structure needs to be done before constructing and putting the vessel into use.??A database is an assembly of meaningful information tabulated in such a way, that we can easily and quickly retrieve the appropriate result. The database management system is a general-purpose software system that promotes the process of defining, constructing and manipulating database for various applications. A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model invented by Edgar F. Codd, of IBM's San Jose Research Laboratory. Most databases in widespread use are based on the relational database model. A relational database is a set of tables containing data fitted into predefined categories. Each table (which is sometimes called a relation) contains one or more data categories in columns. Each row contains a unique instance of data for the categories defined by the columns. For example, a typical business order entry database would include a table that described a customer with columns for name, address, phone number, and so forth. The relational model offers various levels of refinement of the table relations called database normalization. Relational Databases have been providing the storage space support for many decades now with implementations like Oracle, MySQL, etc. [1]. Despite advances in computing, faster processors and high-speed networks, the performance of relational database applications is becoming slower and slower. RDBMS is not the best solution. Database schemas are very strict. When the data gets big, then the traditional SQL join operations may not work. Changes made to a single column will resound and would result in multiple changes. And finally, we need look into the mismatch occurring in different tables due to the changes done. With the intense demand to store huge volumes of information, it was a need to switch from relational databases to graph database.

II. OVERVIEW OF SOFTWARE DEVELOPMENT PROCESSES AND METHODS

2.1. Software Processes

Like most engineering work products where design precedes construction or physical implementation with a properly defined process, the same can be said for software development. The software process is a collection of work activities, actions, and tasks that are performed when some work product (like a software application) is to be created [3]. The generic framework of software processes or phases are communication, planning, modeling, construction, and deployment as shown in Figure 1.

- Communication phase helps to identify a problem that is amenable to a computer-based solution. The overall goals and objectives of the computer-based solution are identified and stated in this phase.
- Planning phase comprises of a set of management and technical practices that defines the road map toward the attainment of the software goals and objectives.
- Modeling phase involves the creation of models that will provide a better understanding of the actual system to be built. The two classes of models that are created in software engineering work are: Requirements or analysis model and Design model. Requirements models are used to represent the requirements of the system identified during the communication and planning phases. Design models represent the characteristics of the system that enable the programmer to construct it effectively. Creating the requirements and design models involves the use of some of the modeling languages or techniques that have been proposed for software documentation. One of such modeling languages is the Unified Modeling Language (UML). The Unified Modeling Language (UML) is the standard language in software engineering for visualizing, specifying, constructing and documenting the artifact of a software-intensive system [4].
- Construction phase encompasses a set of coding and testing tasks that lead to operational software that is ready for delivery to the end user [3].
- Deployment phase consists of the delivery of the work product to the end user, support (such as user manual) for the usage of the product, and feedback from the end user. This phase, often times, takes a number of times as the software moves to completion.

However, during software development, these processes are organized in terms of sequence and time, and this is known as the process flow. The process flow describes how these frameworks of activities, actions and tasks are organized with respect to sequence and time. This organization could be linear, iterative, evolutionary or parallel [3][5]. Shown below is the evolutionary process flow which executes the activities in a circular manner and was chosen as the process model for this application development.

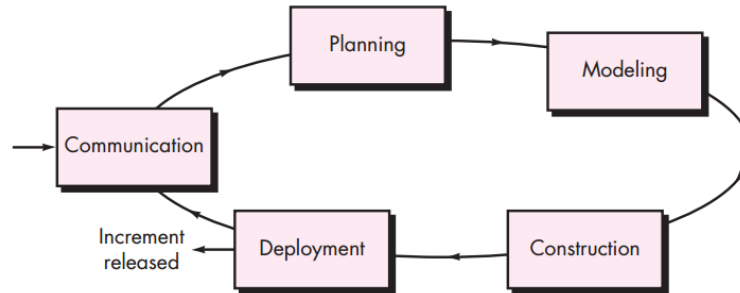


Figure 1: Evolutionary Process Flow [3]

2.2. Software Development Process Methods/Models

Each of the activities, actions, and tasks defined in the software process resides within a framework or model that defines their relationship with the process and with one another. This framework is called the software method or model. Some software process models are the waterfall model, iterative model, incremental model, and the agile model to mention but a few.

The software process models accommodate the described generic framework activities, but each applies a different emphasis to these activities and defines a process flow that invokes each framework activity (as well as software engineering actions and tasks) in a different manner. [3].

For example, the waterfall model emphasizes on the need for the completion of a particular phase before moving on to the next stage. The iterative model emphasizes on the simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed [5]. The incremental model delivers a series of releases, called increments that provide progressively more functionality for the end user as each increment is delivered [3]. While Agile model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product [5].

The Agile model is the preferred model for modern software development [3]. The reason for this preference is because the agile method stresses on the need for a continuous communication and collaboration among developers and customers. Furthermore, the agile method reduces the overheads involved in system development by focusing more on coding rather than on design and using an iterative and incremental approach to software development. Also, the method delivers software that evolves quickly to meet changing requirements. Lastly, the agile method is best suited for small/medium-size business system or PC products [6]. Show in the diagram below (Figure 2) is the graphical illustration of the agile model.

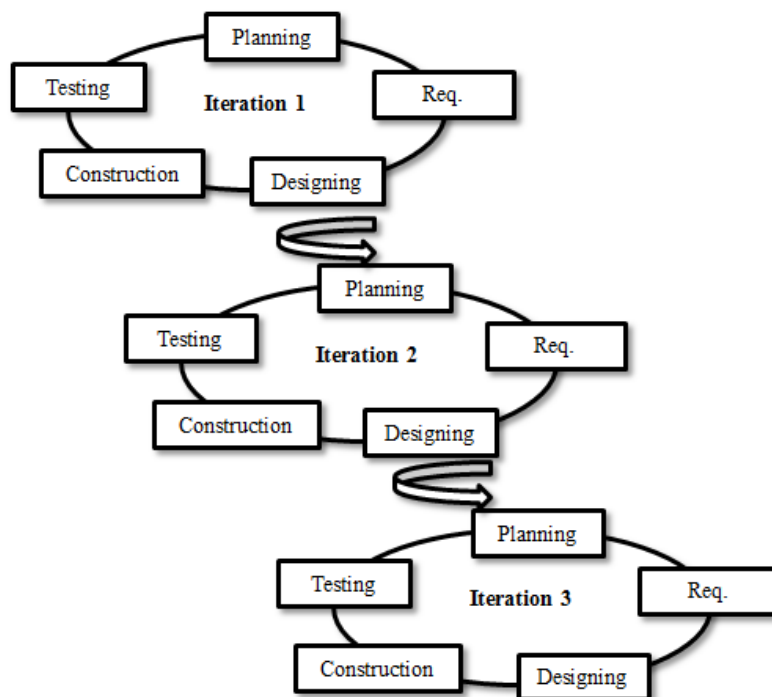


Figure 2: Graphical Illustration of Agile Method

III. METHODOLOGY

The development of the PCA software application utilizes the agile process model because of its inherent advantages and its suitability to the system design and implementation. Using this model, the system evolved from an incomplete system (i.e., one that does not meet all the user's identified requirements) in the first iteration to a complete system (i.e., one that meets the entire user's identified requirements) in the third iteration.

The first, second and third iterations followed all the phases of software development and led to the production of the first, second and third deliveries of the system. While the first and second deliveries were throw-away prototypes, the third delivery was a complete functional system. The processes of the third iteration of the system design, which are refinements and improvements of the first and second iterations, are elaborately discussed in this report.

3.1 Requirements Modeling

Gaining a better understanding of the requirements gathered requires using appropriate requirement models. The use case, data model and Data Flow Diagram (DFD) are used to model the gathered requirements. This is because these models are sufficient for a complete modeling of the requirement of the system.

3.1.1. Use Case Model

The use case diagram represents the usage and functions of the system from the user's view point. The identified use cases from the requirements gathered are depicted in the UML Use Case diagram in Figure 3 below. The description of the various use cases as shown in the diagram is presented in Table 1.

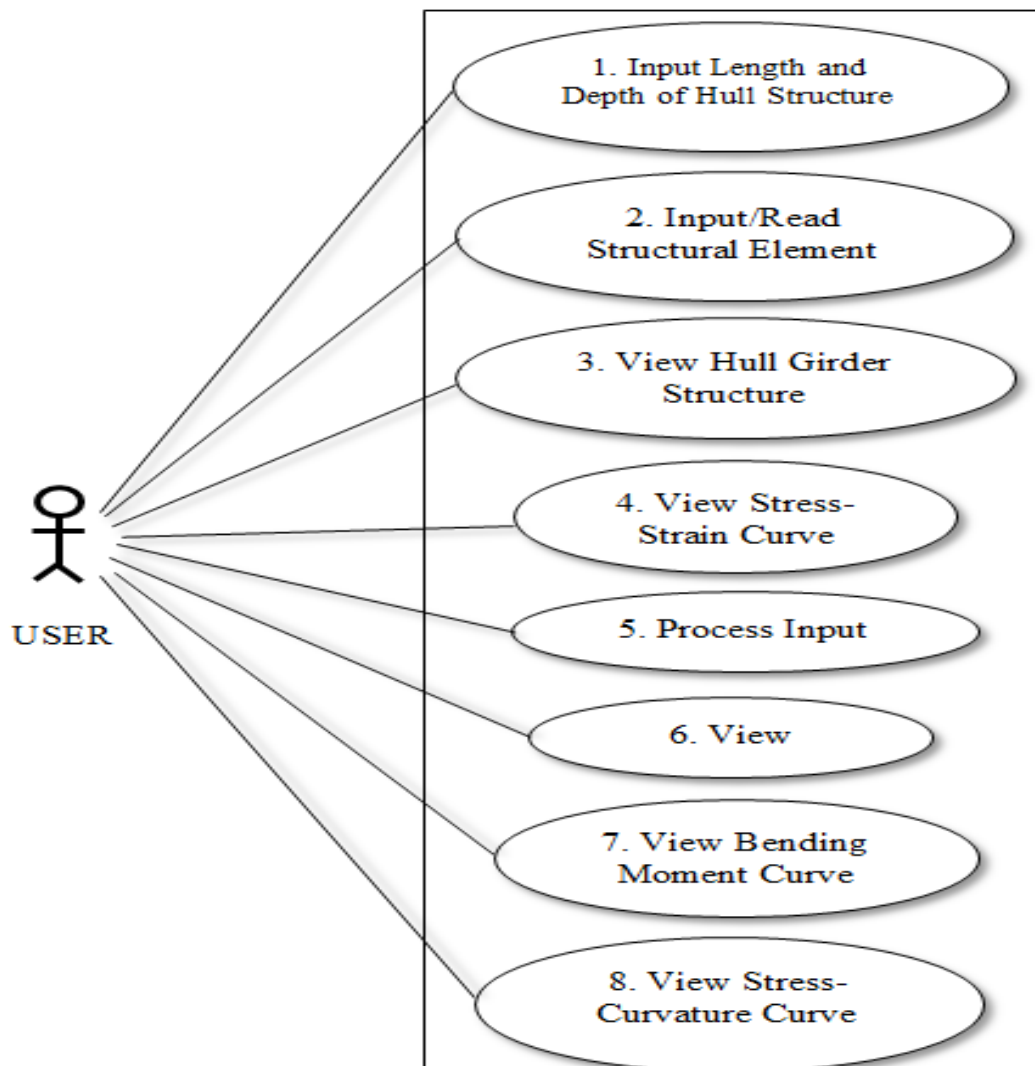


Figure 3: UML Use Case Diagram.

S/n	Input	Purpose	Result	Exception
1	Length and depth	Receive the length and depth of hull structure	Length and depth are stored in the system	Display error message and store no data if: (i) invalid length and depth,(ii) no length and depth input is provided
2	Structural element properties like element type, angle, height, width, and yield stress	Receive the structural element data of the hull structure	Calculation of other element properties like area, centroid and moment of inertia is done and the data are stored in the database along with the computed data.	Display error message if: (i) invalid or no data is entered, (ii) read data from file, (iii) no file or an invalid file is selected
3	Depth, length and element data	Display the cross-section of the hull girder structure	Display the transverse cross-section of the hull girder using colours to indicate element types.	Display the hull cross-section only when 1 and 2 is implemented
4	Button Click, number of curves, element number	Display the stress-strain curve	A plot of the stress-strain curve(s).	If 1 and 2 have not been implemented, no plot will be displayed. If no or invalid entry is made no graph will be displayed
5	Button Click	Process the input data using the formulations	The input data are processed. Additional data like bending moment, Curvature, plastic moment of inertia are generated	No processing will be done if 1 and 2 have not been implemented
6	Button Click	Display the output of the data processing	initial vertical neutral axis, initial moment of inertia, vertical yield bending moment, vertical plastic bending moment of the hull girder structure are displayed in a tabular form	No result will be displayed if the 1,2, and 5 have not been implemented
7	Button Click	To display bending moment curve	Display the bending moment-curvature plot	If 1, 2, and 6 are not implemented, no display. The bending moment of each element can be displayed using the element position.
8	Button Click, the number of curves and the element number (or position)	To display the stress-curvature curve	A plot of the stress-strain curve(s)	No curve will be displayed if: (i) 1, 2 and 5 are not implemented. (ii) No or invalid entry is made

3.1.2 Data Model

The data model provides a representation of the structure of data objects. This model became imperative because the requirements included the need to interface the application with a database.

Entity Relationship Diagram and Data Dictionary are used to model database data objects. However, there is no identified relationship between entities hence; the ER Diagram is not used. Rather, the Data Dictionary is sufficient for the data modeling and is used to depict the data model of the system as shown in Table 2.

- **Activity Log** table keeps track of the elements and their types as they are entered by the user.
- **BM Data** entity stores the computed bending moments and their corresponding curvature increment.
- **Element Stress** table stores the computed stress of each element for every curvature increment.
- **Model Properties** entity stores the properties of the model like Initial Vertical Neutral Axis, Initial Moment of Inertia, Vertical Yield Bending Moment, Vertical Plastic Bending Moment and their corresponding values.
- **Section Properties** table stores the properties of the structural elements (like section type, start points, angle, yield stress, breadth, height, thickness, area, centroid, moment of inertia, minimum yield stress) and their corresponding values.

3.1.3. Data Flow Diagram

A visualization of the flow of data in and out of the system is essential for a better understanding, design and implementation of the requirements of the system. This visualization is represented by the Data flow diagram (DFD). The DFD shows the input-process-output view of a system [3]. The context level DFD of the system showing the transformation of inputs to outputs is depicted in the Figure 4 below.

Table 2: Data Dictionary

DATA DICTIONARY				
ENTITIES/TABLES	FIELDS	DATA TYPE	SIZE	DESCRIPTION
ActivityLog				
	ID	INT	10	PK, AUTO_INCREMENT
	Section	VARCHAR	50	
	SectionElementType	VARCHAR	50	
BMData				
	ID	INT	10	PK, AUTO_INCREMENT
	BM	DOUBLE		
	K	DOUBLE		
ElementStress				
	ID	INT	10	PK, AUTO_INCREMENT
	Element1...N	DOUBLE		
ModelProperties				
	ID	INT	10	PK, AUTO_INCREMENT
	ModelProperties	VARCHAR	50	
	ModelPropertiesValues	DOUBLE		
SectionPropertiesTable				
	ID	INT	10	PK, AUTO_INCREMENT
	SectionType	VARCHAR	50	
	ElementType	VARCHAR	50	
	Angle	DOUBLE		
	ElementSpacing	DOUBLE		
	xStartPoint	DOUBLE		
	yStartPoint	DOUBLE		
	PlateYieldStress	DOUBLE		
	StiffenerYieldStress	DOUBLE		
	PlateThickness	DOUBLE		
	PlateBreadth	DOUBLE		
	WebThickness	DOUBLE		
	WebHeight	DOUBLE		
	FlangeThickness	DOUBLE		
	FlangeBreadth	DOUBLE		
	Centroid	DOUBLE		
	Area	DOUBLE		
	MomentOfInertia	DOUBLE		
	MinimumYieldStress	DOUBLE		

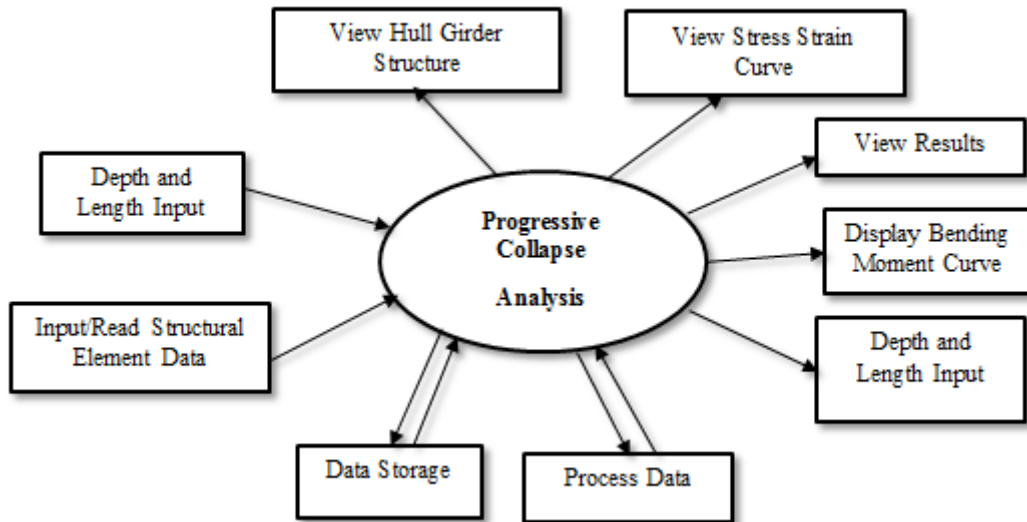


Figure 4: Data Flow Diagram

3.2 System Design Modeling

The design model sets the stage for the construction of the software. It provides detail about software architecture, data structures, interfaces, and components that are necessary to implement the system [3]. The design model for this system has been aptly captured by the Architecture model, the User Interface (UI) Model, the Class Model, and Component Level Model (Flow Chart).

3.2.1. Software Architecture Model.

It models the structure of the system and the manner in which the data and components interact or collaborate with one another [3]. The architectural style chosen for the system architecture is the data-flow architecture as depicted in Figure 5. The architecture is applied when input data are to be transformed through a series of computational components into output data as it is the case with the PCA System. Other architectural styles are the call and return architecture and the layered architecture [3].

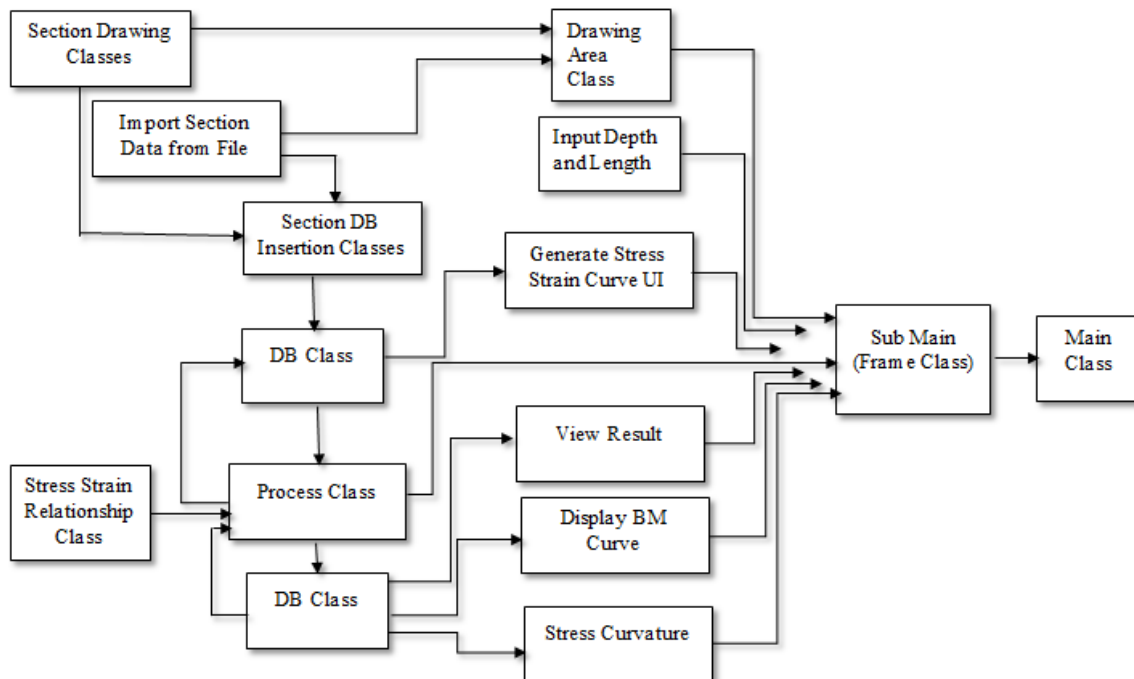


Figure 5: System Architecture

3.2.2. Class Design

Because the requirements emphasize on the usage of object-oriented approach in carrying out the system design, it becomes imperative to present the class model of the system. The class model represents the identified classes of the system and their inter-relationship. The classes and the UML class diagrams that are necessary for the implementation of the system are presented below.

- **Import Data Class:** this class contains methods for reading the hull structure data from an excel file stored in any of the directories in the user’s computer. Shown below in Figure 6 is the UML Class Diagram of the class.

```

ImportDataClass
filePath : String
importDataClass()
getFilePath() : String
readDataFromFile(filePath : String) : String[][]
    
```

Figure 6: Import Data UML Class Diagram

- **Database class:** this class is used to establish a connection with the database, create the tables identified in the data dictionary, insert data, select data and delete data in the database. The UML class diagram of this class is shown below in Figure 7.

```

DatabaseClass
URL: String
connection: Connection
resultSet: ResultSet

DatabaseClass()
createActivityLogTable(): void
createBMCurveDataTable(): void
createElementStressTable(): void
createModelPropertiesTable(): void
createSectionPropertiesTable(): void
dropDBTables(): void
insertIntoActivityLogTable(sectionName : String,sectionElementTypeCuve : String): void
insertIntoBMCurveDataTable(bendingMoment : ArrayList<Double>, curvature : ArrayList<Double>): void
insertIntoElementStressTable(data : double[][]): void
insertIntoModelPropertiesTable(properties : String[], values : String) : void
insertIntoSectionPropertiesTable(sectionType : String, elementType : String, angle : double, elementSpacing : double, xStartPoint : double, yStartPoint : double, plateYieldStress : double, stiffenerYieldStress : double, plateBreadth : double, plateThickness : double, webHeight : double, webThickness : double, flangeBreadth : double, flangeThickness : double, centroid : double, area : double, momentOfInertia : double, minimumStress : double) : void
selectAllSectionData() : String[][]
selectBMDData() : ArrayList<Double>
selectElementStress() : String[][]
selectKData() : ArrayList<Double>
selectModelPropertiesData() : String[][]
selectSectionsFromActivityLog() : ArrayList<String>
    
```

Figure 7: Database Class UML Class Diagram

- **Stress-Strain Relationship Class:** this class computes and produces the stress and strain for the different elements identified in the user’s input data (which is the structural elements of the hull girder). The UML class diagram of this class is depicted Figure 8 below.

```

StressStrainRelationshipClass
modulusOfElasticity : double
lengthOfSection : double
depthOfSection : double

StressStrainRelationshipClass(lengthOfSection : double, depthOfSection : double)
getEdgeFunction(strain : double) : double
getMin(a : double, b : double, c : double) : double
computeStressForHardCornerElementType(bv : double, tv : double, bh : double, th : double, sigmaP : double, sigmaS : double, relAvStn : double) : double
computeStressForPlateElementType(b : double, t : double, sigmaP, : double, relAvStn : double) :
    
```

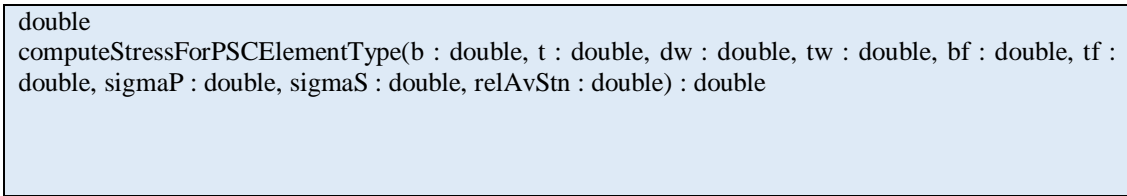



Figure 8: Stress/Strain Relationship UML class diagram

- **Data Processing (Computation) Class:** this inherits some properties from the SwingWorker Class. It uses the inputs provided by the user, processes them using derived formulations and sends the results of the processing to the database. The UML class diagram is shown in Figure 9 below.

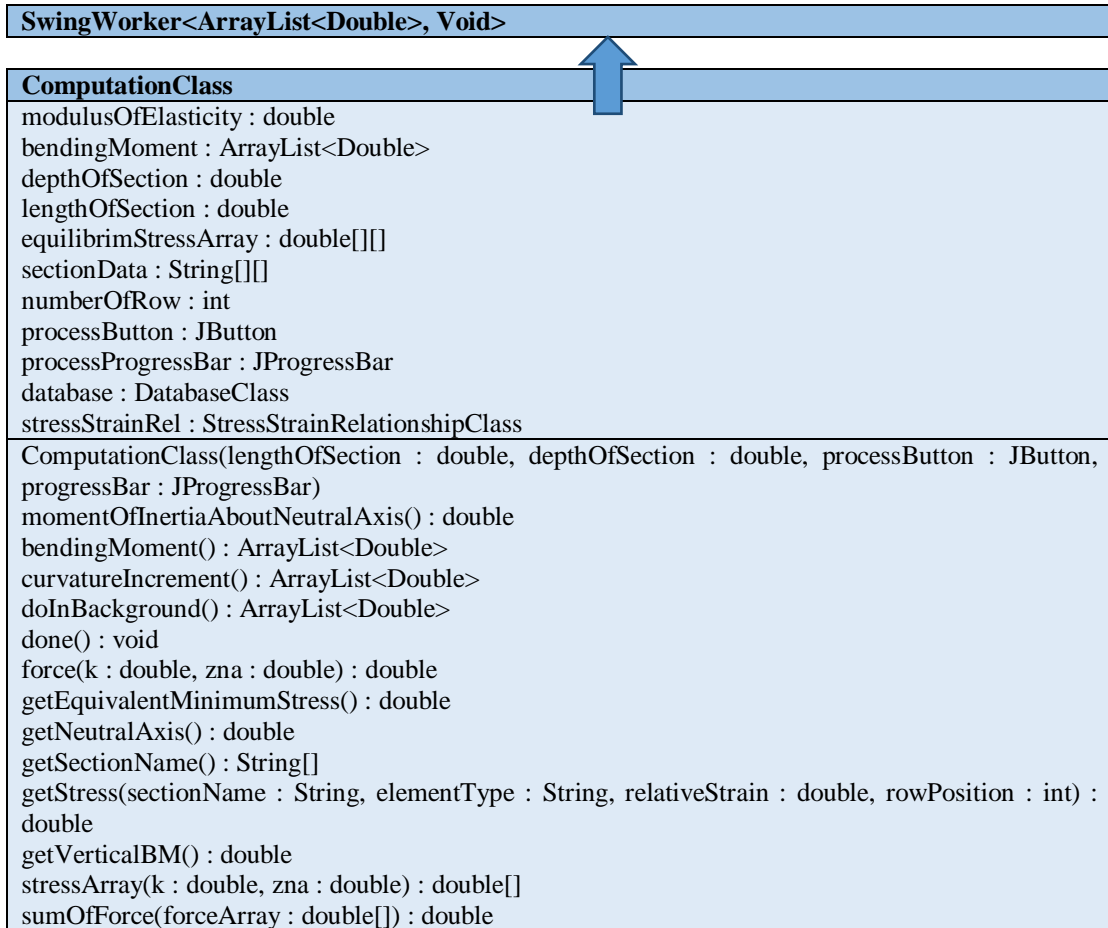


Figure 9: Computation Class UML class diagram

- **Display Bending Moment Class:** this class displays a plot of the bending moment against curvature. Both data are obtained from the database. The UML class diagram is shown Figure 10.

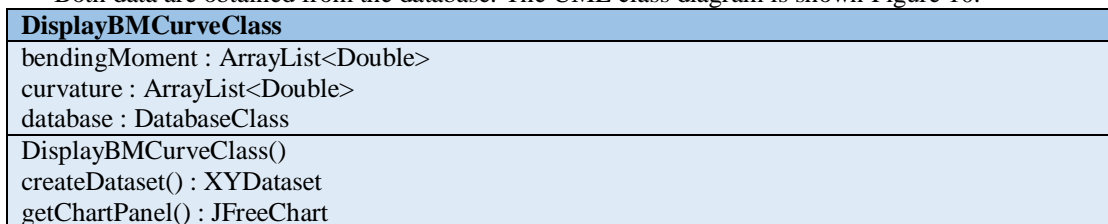


Figure 10: Display BM Curve UML class diagram

3.2.3. User Interface (UI) Design.

This design is aimed at creating an effective communication medium between the user of the system and the system itself. The UI of the system has been designed taking into consideration the use case scenarios

depicted in the requirement model. Multithreading is as well incorporated in the design process to ensure that the GUI remains responsive to user interaction especially during lengthy data processing. A prototype of the user interface showing system component layout, menus and icons is shown in the design below (Figure 11). This prototype has been designed using MS Power Point Storyboarding tool. Also, a progress bar indicating the progress of the processing task to be carried out by the system is designed as part of the UI.

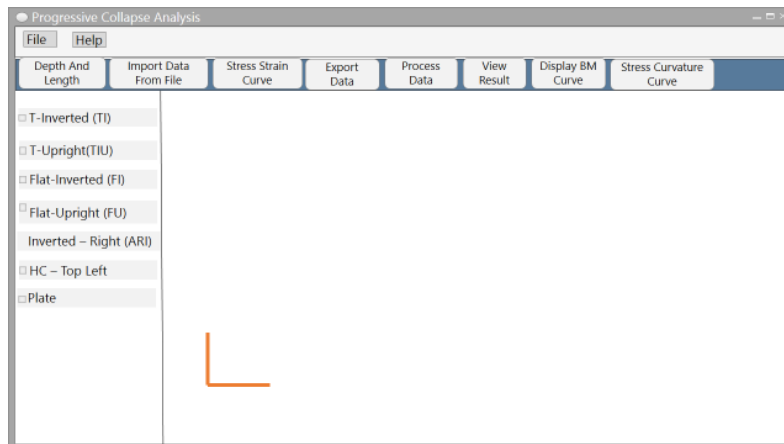


Figure 11: System UI Prototype.

3.2.4. Component Level Design.

One important model used in the component level design is the flow chart. The system flow chart is the pictorial representation of the logical sequence through which data is supposed to flow from the end user to the system and back to the end user. Shown below in Figure 12 is the system Flow Chart utilized in this work.

3.3 Software Development

The implementation or construction (i.e. coding) of the system design is done using different technologies and tools. Java, an object-oriented programming language, is used for the construction or coding of the classes designed. The reason for this is because Java is simple, object oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high performance, multithreaded, and dynamic [8]. Also, Java has a large collection of libraries known as Application Programming Interfaces (APIs). It should also be pointed out that Java is write one, run anywhere (i.e. platform independent) [9] and as such programs written in java can be executed on any PC having any operating system installed. SQL was used to create the database tables identified in the data dictionary. The insertion, retrieval and deletion of data were achieved using SQL.

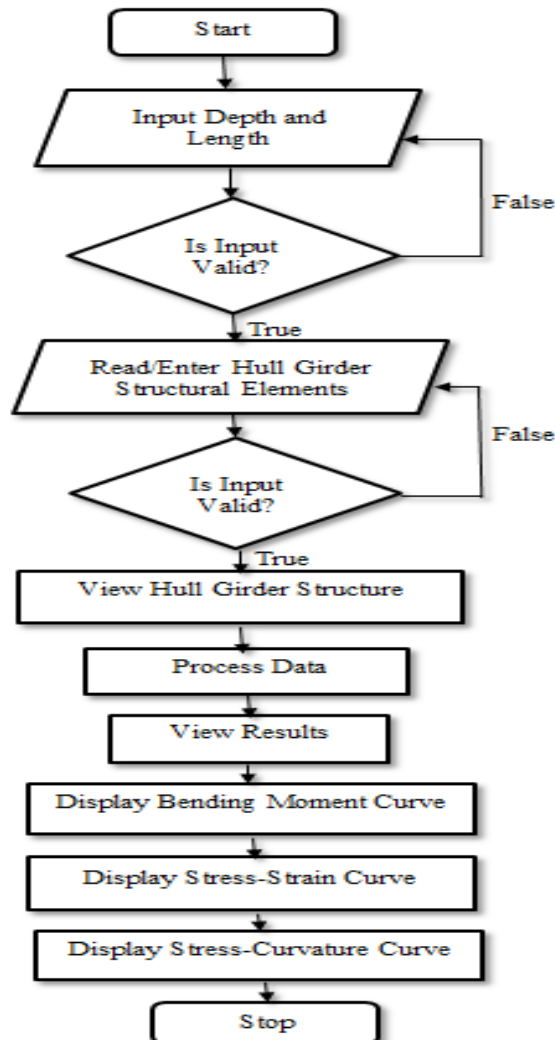


Figure 12: System Flow Chart

3.3.1. Tools Used

➤ NetBeans IDE

NetBeans Integrated Development Environment (IDE) is used for writing all of the Java codes for the classes designed. It supports features like error checking and detection, code completion and conversion of source files to executable jar files to mention but a few, and these make the development of codes easy and less frustrating.

➤ Advanced Installer.

The Advance Installer, a software application, makes it easy to convert an executable jar file to an installable file. It was used to convert the jar file generated with Netbeans IDE to a set up file that can be installed on the user's PC.

➤ H2 database.

The database is a relational database management system (RDMS) that uses SQL for the manipulation of data stored in it. The database was used for storage of user's input data and was embedded with the PCA system

3.3.2. Libraries (APIs) Used.

As stated earlier, Java has a large collection of libraries which encourages code reusability. These libraries are referred to as Application programming Interfaces (APIs). For the construction of the PCA system, some of these APIs were used. Among the APIs used are:

- i. JExcel API for reading and sending data to excel files.
- ii. JFreechart API for producing the plots required by system
- iii. JAMA (Java Matrix) API for carrying out the needed matrix operations
- iv. H2 Database API for establishing connection to the H2 embedded in the application.
- v. JTattoo API for enhancing the look and feel of the application User Interface.

IV. RESULTS AND DISCUSSION

The results obtained in the course of developing the Progressive Collapse Analysis (PCA) system are products of the software testing phase, which is a part of the software development process. In order to ensure that the system conforms to the stated objectives and meet the user's requirements, the PCA System was tested to determine its responses to the following:

- i. Authentication of user inputs (i.e. length and depth and structural elements of the hull girder structure) to ensure that valid inputs are entered into the system.
- ii. Storage and maintenance of data in the database
- iii. Processing of user's inputs using the derived formulations
- iv. Generation of outputs in the form of displaying the model properties, bending moment curve, stress-strain curve and stress-curvature curve.

➤ **Input Depth and Length of hull structure.**

The result for this component of the program shows a dialog box used for the reception of input from the user (Figure 13). The input is the length and depth of the hull structure.

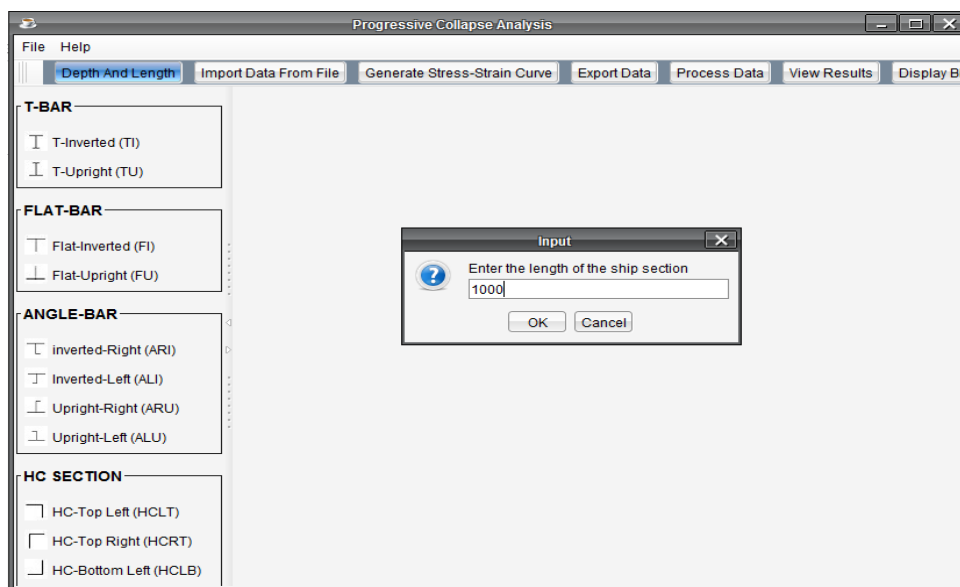


Figure 13: Screenshot for Input of Depth and Length of Structure

An error message is displayed when no or invalid entry is made as shown below in Figure 14. The system stores the input only when it is valid.

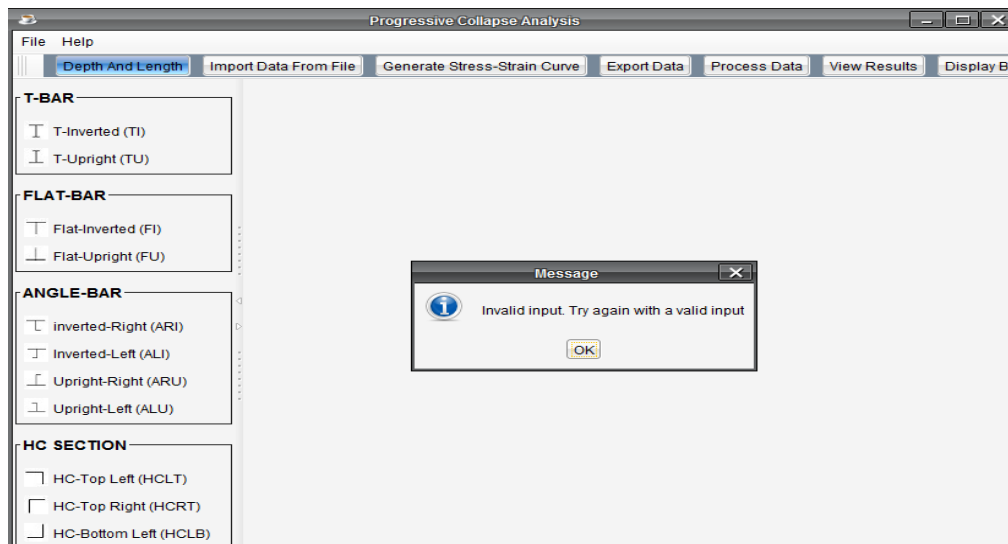


Figure 14: screenshot for authentication user inputs

➤ **Reading Structural Element Data From file and Viewing the geometry of Structure**

A dialog box is displayed for the user to browse to the location in his PC where the file containing the structural element data is stored as shown in Figure 15.

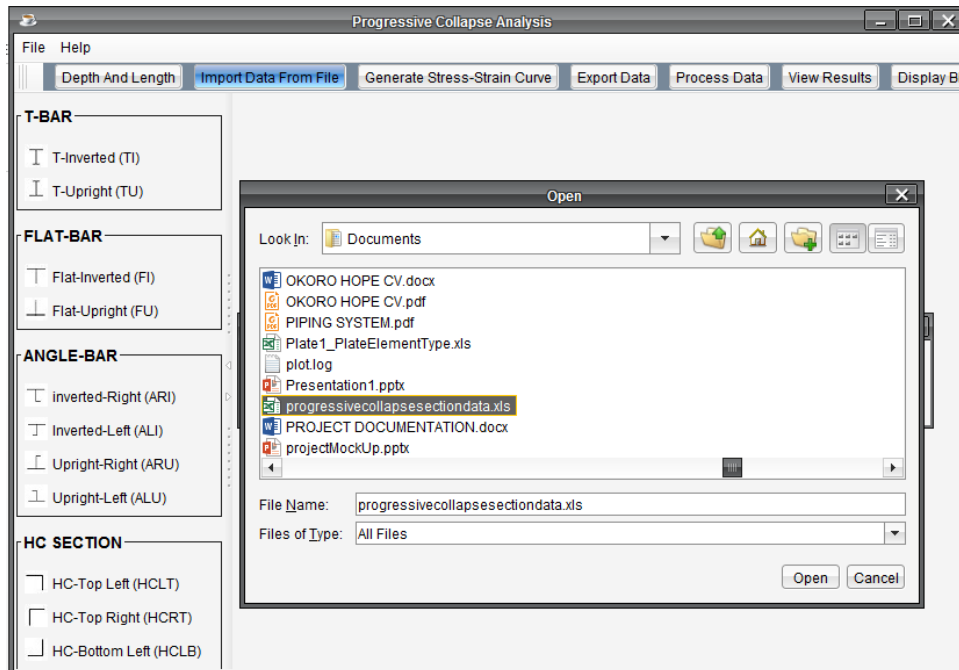


Figure 15: screenshot for importing structural data from file

An error message like the one displayed in Figure 14 above is displayed if no or invalid file was selected.

On the other hand, if a valid excel file is selected and the file contains data in the prescribed format (see appendix A), the system populates the database with these data from the file and displays the geometry of the structure as shown in the Figure 16 below.

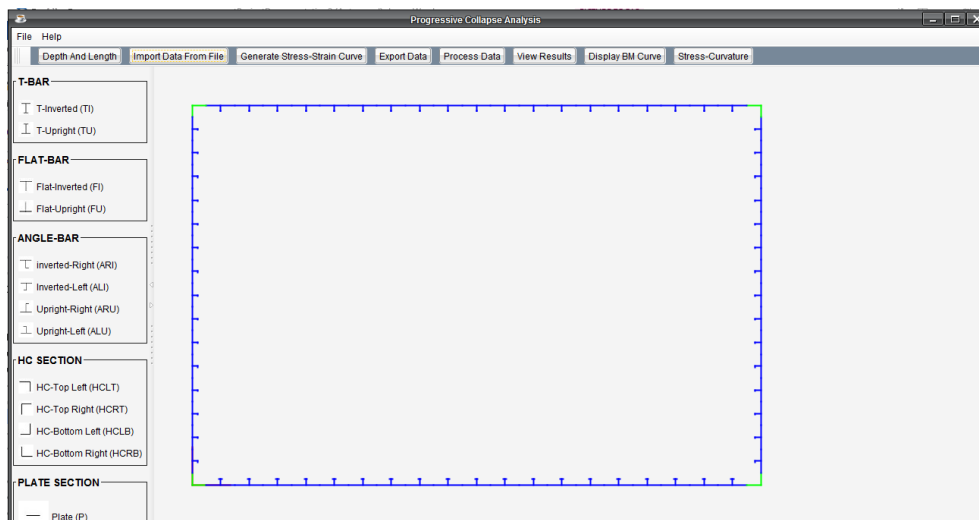


Figure 16: screenshot for viewing the shape or geometry of the structure

➤ **Processing Of Input Data**

The result of this test shows the progress of the computation and processing of the inputted data as indicated by the progress bar shown in the figure below. It also indicates the responsiveness of the system as the processing is ongoing.

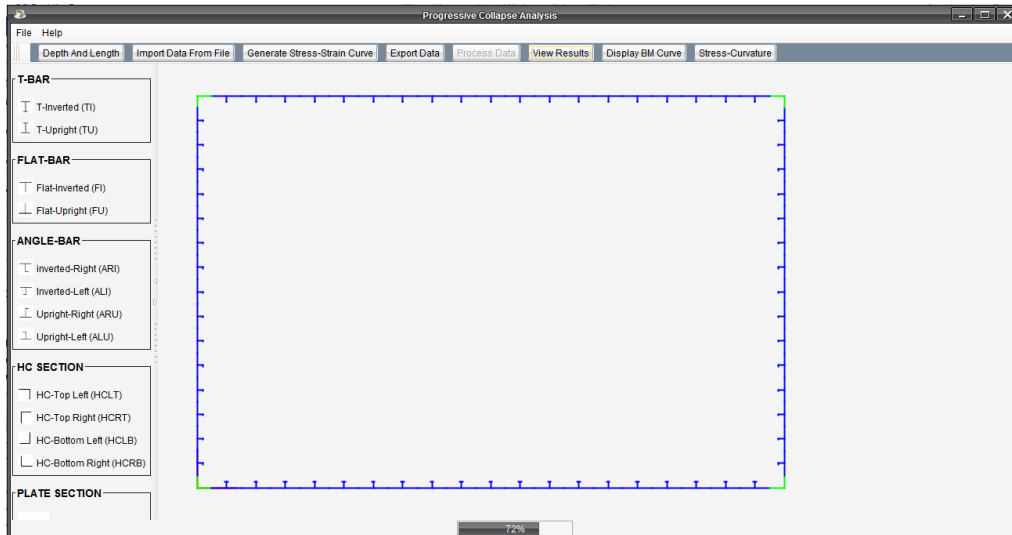


Figure 17: screenshot for processing of input data

➤ **Storage and maintenance of data.**

The test result shows both user's input and the processed data that are stored and maintained by the PCA System using the embedded database. The data are stored in a tabular format in the database by the system. Figure 18 below shows a screenshot of the model properties table test result. This test result displays in tabular format the results obtained from data processing.

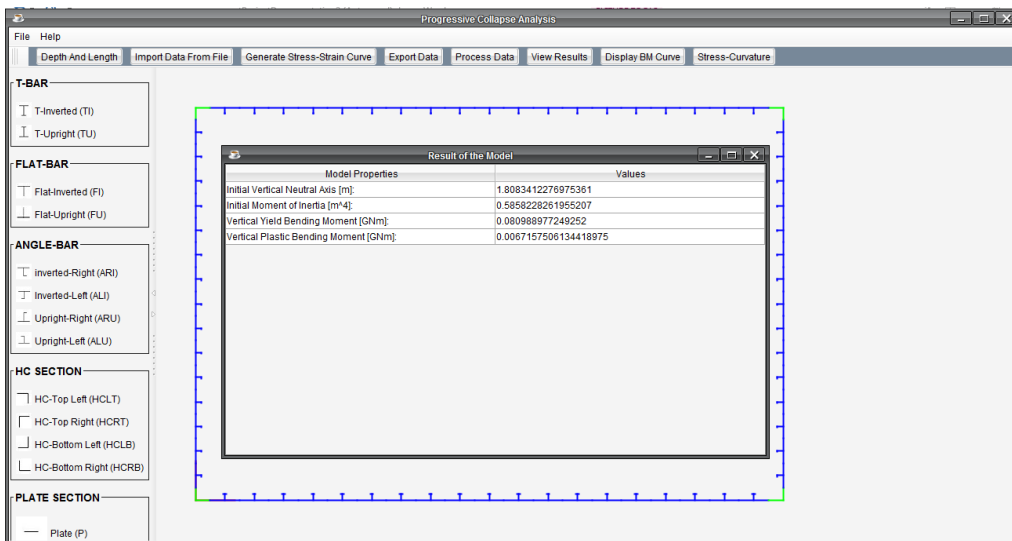


Figure 18: screenshot for viewing model properties

➤ **Display Bending Moment Curve**

This test result displays the bending moment plot of the structure.

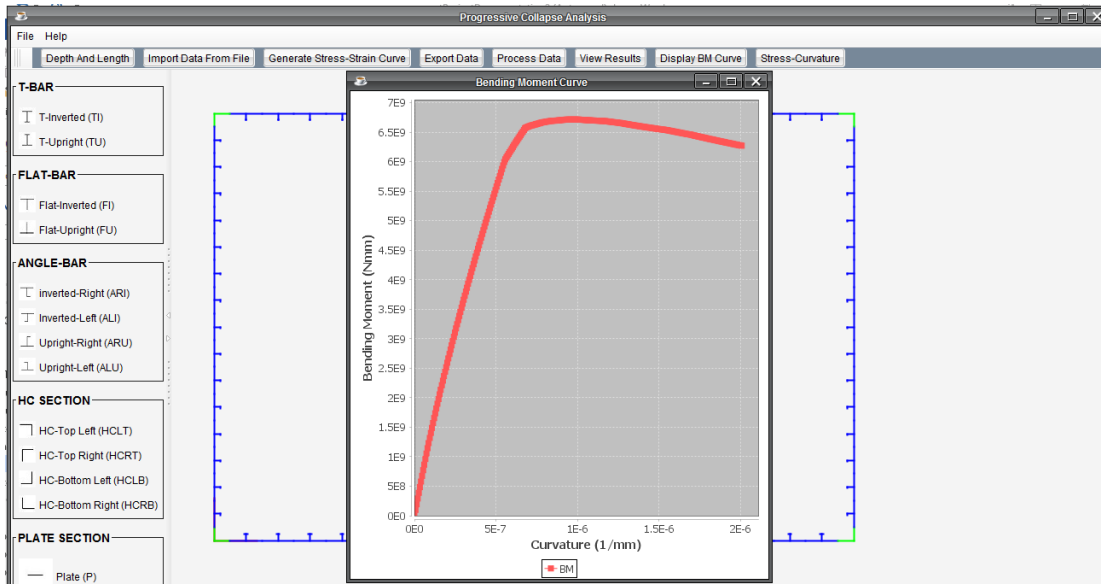


Figure 19: screenshot for viewing the bending moment curve.

➤ **Display Stress Curvature Curve**

The stress curvature curve(s) for the desired elements that made up the structure is or are displayed. The test result shows the stress strain curve(s) for the structural elements desired by the user.

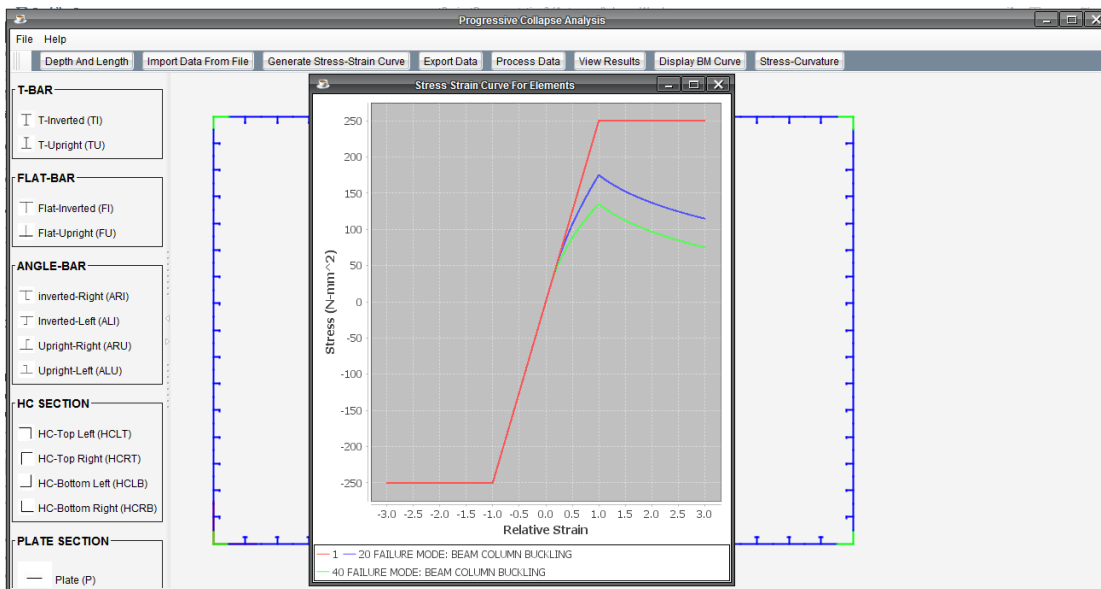


Figure 20: screenshot for viewing stress-strain curve

The results which are a product of software testing ensured that errors in the system were uncovered and corrected. And because of the agile methodology used, each iteration provided an avenue for refining the requirements and fixing the bugs in the code. This produced a system that is able to meet the stipulated objectives and requirements at end of the third iteration of the design process.

The software is deployed as a functional desktop application. It can be installed on the user's computer and executed by double clicking on the desktop short-cut created. Also, a user's manual indicating the usage of the application is deployed with the work product.

V. CONCLUSION

The Progressive Collapse Analysis (PCA) System assists in computerizing the processes involved in analyzing a given ship hull girder structure. The analysis of the structure using the PCA System is important to the safety and longevity of a vessel. The PCA System determines the maximum loading condition that the structure can withstand and the stress-strain relationship of elements or members that make up the structure.

Following standardized processes and modern methods (Agile method) for software development, the PCA System has been designed and tested to provide these functionalities with great ease in usage, acceptable accuracy of results, and great speed in data processing. This system will in no small measure be invaluable to stakeholders and professionals in the field of engineering in general and those in marine engineering related fields in particular.

REFERENCE

References

- [1]. I.J. Hope., "Progressive Collapse Analysis of Ship Hull Girder." M.Sc. thesis, Rivers State University of Science and Technology, Nigeria, 2016.
- [2]. Hughes et al. in 2010. Help include this reference from your work. Thanks.
- [3]. S. P. Rogers. Software Engineering, A Practitioner's Approach. New York: McGraw Hill, 2010.
- [4]. G. Booch, J. Rumbaugh, and I. Jacobsen. The Unified Modeling Language User Guide. Boston: Addison-Wesley, 2005.
- [5]. Tutorialspoint. "Software Development Life Cycle." Internet: http://www.tutorialspoint.com/sdlc_tutorial.pdf, [Nov. 29, 2016].
- [6]. G, Chukwudebe. EEE 531. Class Lecture, Topic: "Agile Software Development." EN 1, Faculty of Engineering, Niger Delta University, May. 11, 2014.
- [7]. M. Kifer, A. Bernstein and P.M. Lewis. Database Systems, An Application-Oriented Approach. New York: Pearson Education, 2005.
- [8]. Y.D. Liang. Introduction to Java Programming. New Jersey: Prentice Hall, 2011.
- [9]. Oracle. "The Java Tutorials." Internet: <https://docs.oracle.com/javase/tutorial/getStarted/intro/index.html>, [Dec. 06, 2016].

Ikue-John, Hope, et. al. "Design and Implementation of a Software Application for Progressive Collapse Analysis of Ship Hull Girder Structures." *The International Journal of Engineering and Science (IJES)*, 10(07), (2021): pp. 25-40.