**University of Jordan**

**School of Engineering**

**Computer Engineering Department**

**GP01 – Graduation Project Proposal**

|  |  |  |  |
| --- | --- | --- | --- |
| *Fall/2022* | **Semester / Year** | **Fahed Jubair** | **Supervisor** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Signature** | **GPA** | **Dept.** | **ID Number** | **Student Name** | **No.** |
|  | **3.66** | Computer Engineering | 0186436 | Ammar Abu Yaman | **2** |
|  | **3.89** | Computer Engineering | 0186435 | Feras AllNajjar |

**Emails (To contact you - if needed - after graduation):**

**Student 1: ammar.abu.yaman@gmail.com  
Student 2:**

**Choose all that applies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Type** | **Hardware** | **Software / App / Web Development** | **Research** |

Please completely fill all required fields below. Do not change fonts or font size. **The form should not exceed four pages.**

|  |  |  |
| --- | --- | --- |
| **Title of Senior Year Project** | Finite State Machine to Java source to source compiler | |
| **Project Summary** | The Finite state machine (FSM) is a model of computations, it can be in one of a finite number of states at a given time and can transition from one state to another based on the current state and with the user’s input.  It’s a very useful tool to model many computational problems in computer science, mathematics, biology …etc. but they are usually limited to the design phase as a guidance to the implementation. The State design pattern can be used to efficiently implement FSM but it is very tedious to write and causes the business logic to be scattered throughout the codebase and becomes harder to make sense of it[1]. Some tools exist that can convert textual representation of a FSM into working code that can be incorporated into a project. But these tools are usually limited into a text format and are often domain specific and have complicated licensing. We propose to build an open source and extensible general purpose tool where a user can create finite state machine to model their algorithm via a friendly graphical user interface and an c that can convert the graphical FSM into working Java code. This will allow this tool to be used anywhere where FSMs are applicable and draw greater accessibly to the use of FSM in designs through the user-friendly GUI with users being able to see the various states and transitions in a visual manner.  **Methodology**:   * Create a web based user Interface where users can create states, specify transitions and actions in a visual drag & and drop style. * Convert the visual FSM into a portable format (XML or JSON) and send it to the backend. * Use the backend engine to compile the FSM into a set of Java classes. representing the states and transitions. * Allow the user to download the generated classes where they can directly incorporate into their project.   **Objectives**:   * Provide an open source and extensible visual tool where people can use to design and implement FSM algorithms and visualize, all in one place with less effort and cost. * Increase the accessibility of FSM and help people adopt them and incorporate them more easily in their projects through user friendly design.   [1] Gamma Erich, Helm Richard, Johnson Ralph, Vlissides John,  Design Patterns: Elements of Reusable Object-Oriented Software, 1994. | |
| **Project Impact** | Creating software is hard and expensive and takes weeks to months from designing to prototyping to implementation, it is estimated that new software can cost between 25K to 250K with months of development time. Often automation tools that can aid in the design and implementation of software can cut down on development time and cost significantly. Our tool aims to help developers express algorithms that can be naturally modeled as a FSM and streamline the process of implementation where our tool takes care of the tedious and error prone code and makes it easy to change and extend the design without spending resources on reimplementing it as our tool takes care of that part. | |
| **Engineering Standards to be used (if any)** | None | |
| * **Simulators** * **Cloud Services** * **Operating Systems** | *Linux, embedded Linux, RTOS, … etc. AWS, Azure, Google Cloud … etc.  ns3 simulator* | |
| * **Software Tools or IDEs** * **Software or Hardware Programming Languages** * **Libraries/Drivers** * **Databases** * **Data Sets** | **IDEs**: IntelliJ,. VSCode.  **Programming Languages**: Java, JavaScript. JSON, XML, HTML, CSS  **Libraries**: Antlr, React | |
| **Project Constraints, if any.** | *None* | |
| **Gantt Chart** | *Kindly attach as a* ***PDF*** *or* ***Excel*** *with this proposal a Gantt chart of the project plan. You learnt about the Gantt chart in Introduction to Engineering course. You must divide your time spent on the project into phases and subphases, have start/end date and expected duration for each part, then list who is going to work on each part* | |
| **Final Deliverables** | A software service that can be used on it’s own via command line interface but with a Web app serving as a visual design tool will encapsulate it to streamline the development process will be delivered. | |
| **Compulsory Deliverables:** | 1. **Project 1 Progress Report** 2. **Final Documentation** 3. **Presentation Slides** |

For graduation project committee use (please do not write below this point):

**Final Committee Decision:**

☐ Approved ☐ Language modifications required  
☐ Approved, with **minor** modifications.  
☐ Approved, with **major** modifications.  
☐ **Rejected**, submit new project idea.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Signature** | **Date** |
| **Chair** |  |  |  |
| **Member I** |  |  |  |
| **Member II** |  |  |  |

**Detailed Committee Members Remarks**

|  |  |  |
| --- | --- | --- |
|  | **Comments** | **Recommendation** |
| **Chair** |  | ☐ Approved  ☐ Approved, with  **minor** modifications  ☐ Approved, with  **major** modifications  ☐ **Rejected**, submit   new project idea. |
| **Member I** |  | ☐ Approved  ☐ Approved, with  **minor** modifications  ☐ Approved, with  **major** modifications  ☐ **Rejected**, submit   new project idea. |
| **Member II** |  | ☐ Approved  ☐ Approved, with  **minor** modifications  ☐ Approved, with  **major** modifications  ☐ **Rejected**, submit   new project idea. |