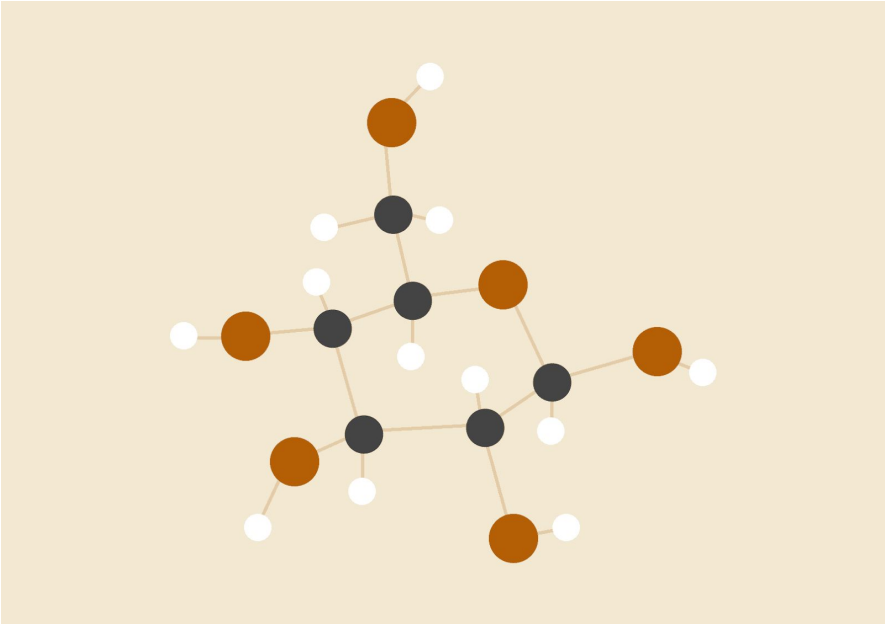
PATENTCONNECT

Software Requirements Specification

Empowering Developers, Designers, Hobbyists, & Alike



DEVELOPERS:

Yousef Hashemi <alhashey@mcmaster.ca>

Jatin Chowdhary <chowdhaj@mcmaster.ca>

Steven Gonder <gondes1@mcmaster.ca>

Yiding Li <liy443@mcmaster.ca>

Varun Verma <vermav6@mcmaster.ca>

LAB: 2 → **GROUP:** 5

The developers of PatentConnect electronically sign & date that the submitted work is strictly our own

The Domain

PatentConnect is a multi-faceted patent landscaping application that enables its clients to efficiently and effectively search for patent citations and relations. It will recurse through the Patent Citation Network and return an engaging collection of information illustrating the connection(s) between patents. PatentConnect is built, using Java, allowing for portability across a wide array of computational devices, namely the personal computer. The goal of PatentConnect is to democratize the patent landscaping industry and empower (potential) developers, designers, inventors, and hobbyists, in a multitude of ways; these are the stakeholders.

PatentConnect will facilitate stakeholders in understanding and interpreting intellectual Property (IP). The main stakeholders of PatentConnect are:

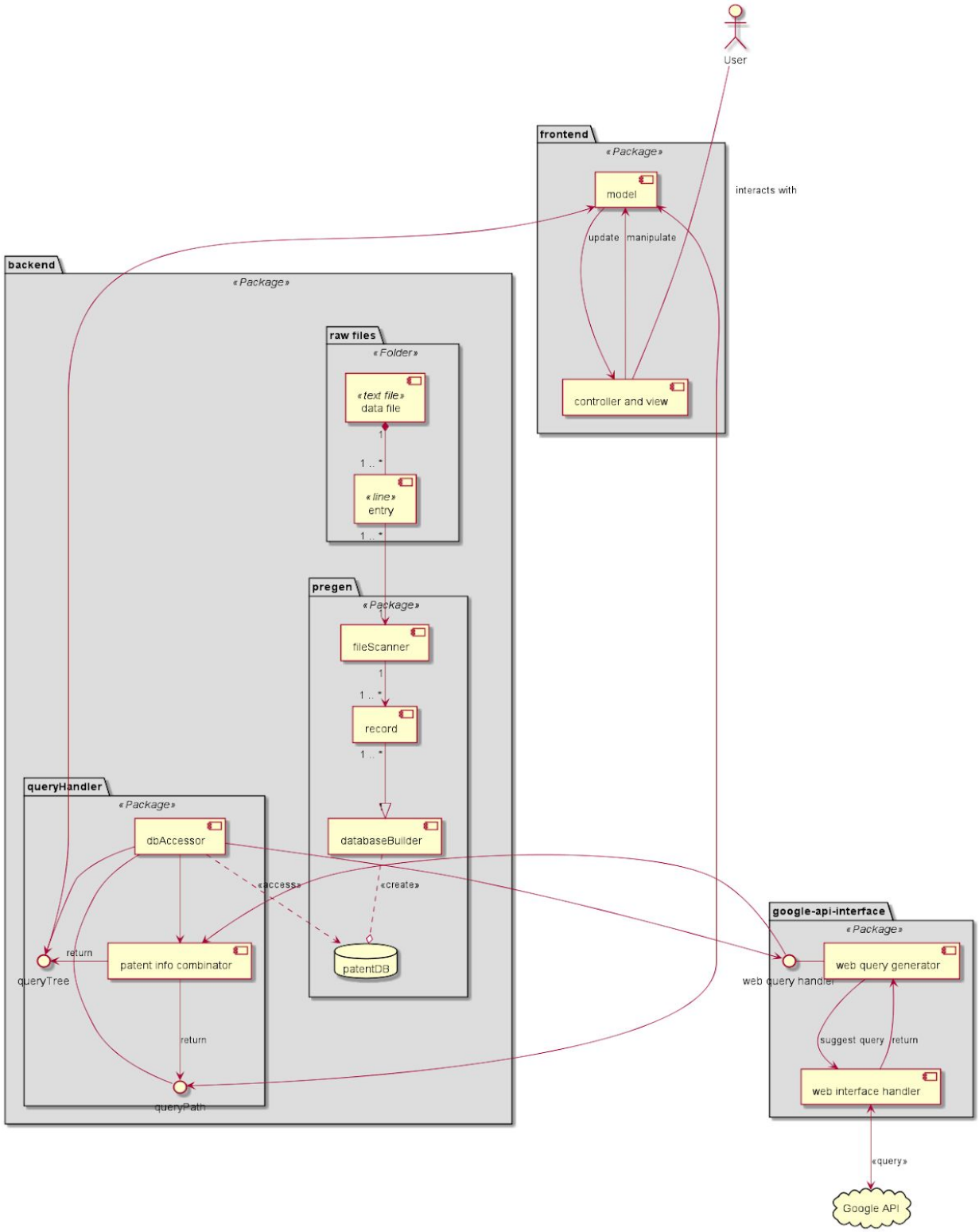
- **Inventors**, which includes developers, designers, and innovators. The inventors are the primary stakeholders of PatentConnect because they are the creators of products.
    - *Goals:* Provide inspiration through previous inventions, and allow inventors to discover how certain patents came to be. This will inspire (potential) inventors to innovate in fields with a plethora of patents.
    - *Expectations:* Inventors want to be able to develop new products without the fear of infringing on existing intellectual property. Thus, they will find PatentConnect to be a crucial tool in the innovative process of invention.
- **Lawyers** and legal teams are an imperative part in combating theft of intellectual property and infringement.
    - *Goals:* Aid and assist lawyers in withstanding frivolous lawsuits brought on by patent trolls.
    - *Expectations:* Lawyers need to be able to effectively convince a judge or a jury of their peers that their client's invention does not impose another product. PatentConnect will help strengthen their case.
- **Consumers** are the primary purchasers of goods and services. Their actions dictate the success of a product and the market.
    - *Goals:* Make patent information more accessible to the general public. People have a right to know how their products are designed and came to be.
    - *Expectations:* In recent years, consumers are becoming more conscious about the products they buy. They want companies to be fair and respectful of the environment, other companies, and customers. Researching relevant information on a product is a tedious process. PatentConnect aims to help minimize that.

The main entities that characterize the domain are the inventors and lawyers; these two are the main cohorts that are actively involved in the patent landscaping progress. These two groups are intertwined and actively collaborate to prevent patent infringement, and fight IP lawsuits. Typically, an inventor creates a patent and the lawyer gets it approved by the patent office. If the inventor is sued for patent infringement, lawyers are hired to contest the accusation. The goal of PatentConnect is to prevent and contest lawsuits by making it easier for inventors to explore patents similar to theirs, and help lawyers create a stronger case, respectively. Simultaneously, PatentConnect aims to inspire inventors to create, draw, build, and innovate by providing an *ideator* platform.

Functional Requirements

This product takes a patent search as the input, and provides the user with a collection of patents related to this patent, and if the user selects one of the patents they will get a description of the patent, as well as links to other patents that the selected patent references. This is achieved by the program maintaining a B-Tree that holds links to files of collections of patents. Whenever a query is made and a patent is selected the program makes a Query to Google Patents for a description of the patent to display. Below is a Unified Modelling Language diagram for PatentConnect.

UML Diagram of PatentConnect



The developers of PatentConnect electronically sign
& date that the submitted work is strictly our own

Non-Functional Requirements

The application will remain accessible at all times while the user is connected to the internet. Maintenance will be done through software updates, pushed via the internet. Patent data pulled from Google Patents and datasets will not be intercepted by third parties; authentication between client and host will be reliably achieved. We will incorporate best practices for security and ensure that our application provides confidentiality through encryption during transfer of information.

The data will not have errors or missing data points. There should be no reason for there to be a margin of error in our results as we are using discrete data. The same result should be computed for every equivalent search the user makes.

Our application will have failsafe properties such that in the event of a non-terminating operation (i.e. infinite recursion, endless loop, etc.), the operation will be forcefully terminated. The operation and termination will not crash or lag the end user's computer. The program will run on most modern household computers with ease.

We will ensure that the end user finds the User Interface (UI) intuitive. Every component of it will be specific and unambiguous. The UI will be essential, minimal, and have thorough documentation for every feature.

Since PatentConnect will be developed using Java, we can count on the JVM's portability properties. Furthermore, it will smoothly work on a variety of operating systems (i.e. Linux, Windows, MacOS, etc.), and will be optimized for lower-end machines.

Requirements On The Development & Maintenance Process

In order to create maintainable code, the development process is governed by a set of guidelines for writing, checking, and testing code. When writing code, the team will adhere to the Google Java Style Guide. This will streamline code reviews and make the overall process of quality assurance much easier. Furthermore, adhering to the Google Java Style will enable the use of automatic style checkers such as Checkstyle.

Code review will be done on a regular basis to ensure quality code is created for the duration of the project. This entails one developer reading through the code, and trying to find issues and places for improvement. This will also give an opportunity for

the developers to double check that the code is meeting the requirements laid out in the *Non-Functional Requirements* section. During code reviews a static code analysis tool such as CodeScan could be used to help detect bugs, dead code, or duplicate code causing bloat in the codebase.

Code testing will be accomplished through the use of the JUnit unit testing framework. Developers will write and run test code for each function created, and the tests will try to cover edge cases, average cases, exceptions, as well as achieving full code coverage.

It is likely that as the project progresses, certain changes will be made to the development and maintenance process. These changes are likely to occur in the tools that are being used. For example, if JUnit is cumbersome to incorporate, then a different testing framework may be used. In addition, if an automatic style checker creates unnecessary overhead, and the developers can format and style code without it, it will no longer be utilized. With the exception of changing tools, frameworks, and other development assistive tools, the process of creating the application will remain consistent with the specification throughout the development cycle.