OSG – Widget Dictionary
CSCE 4925 IT CAPSTONE II

Abstract

This system is designed for web developers, allowing them to use dictionary items that act as a template to create web pages. The system forms a domain-specific-language that requires basic information from the developer to help better interpret their vision and form specified widgets with minimal syntax. There are several templates that comprise the dictionary of available resources the developer can use. These different widgets can interact with PostgreSQL database fields and other specified widgets. The system primarily uses Ruby on Rails to form the necessary Javascript, Hasura and Apollo to form queries to the database and update the information displayed by the page and its widgets.
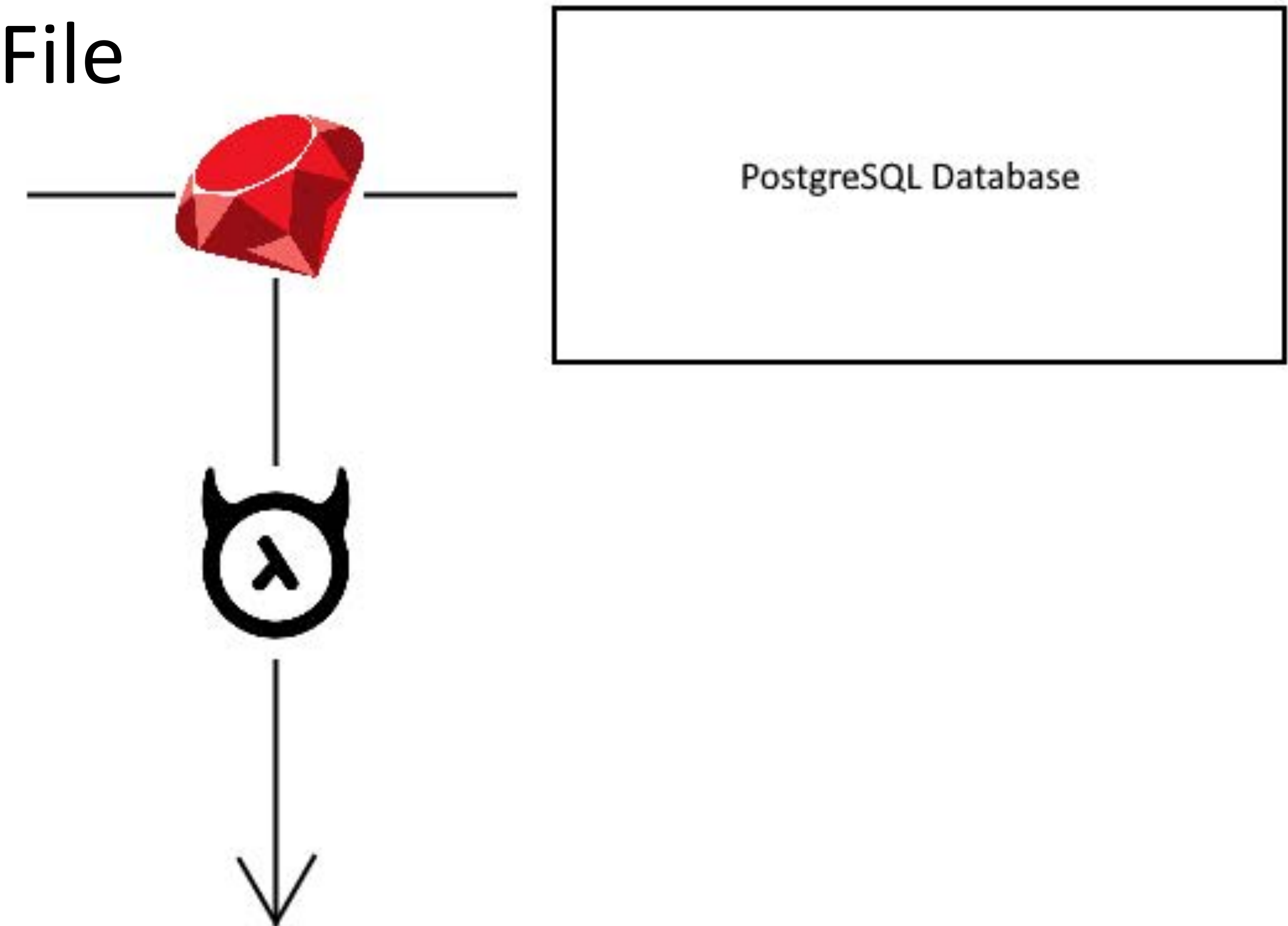
Future Plans

The dictionary can be continuously expanded with new widgets and their respective templates. It acts as a template library.

Work still to Complete

We're currently adding a few more widgets to the dictionary and polishing up the code. We'll be updating the documentation accordingly.

Instructor - David Keathly
Client - Tom Agnew
(Lease Analytics)


```
grid:
   tableName: data
   columns:
      - [id, Int!]
      - [name, String!, dxSelectBox]
      - [description, String!]
      - [balance, money!]
      - [notes, String!, dxTextArea]
```
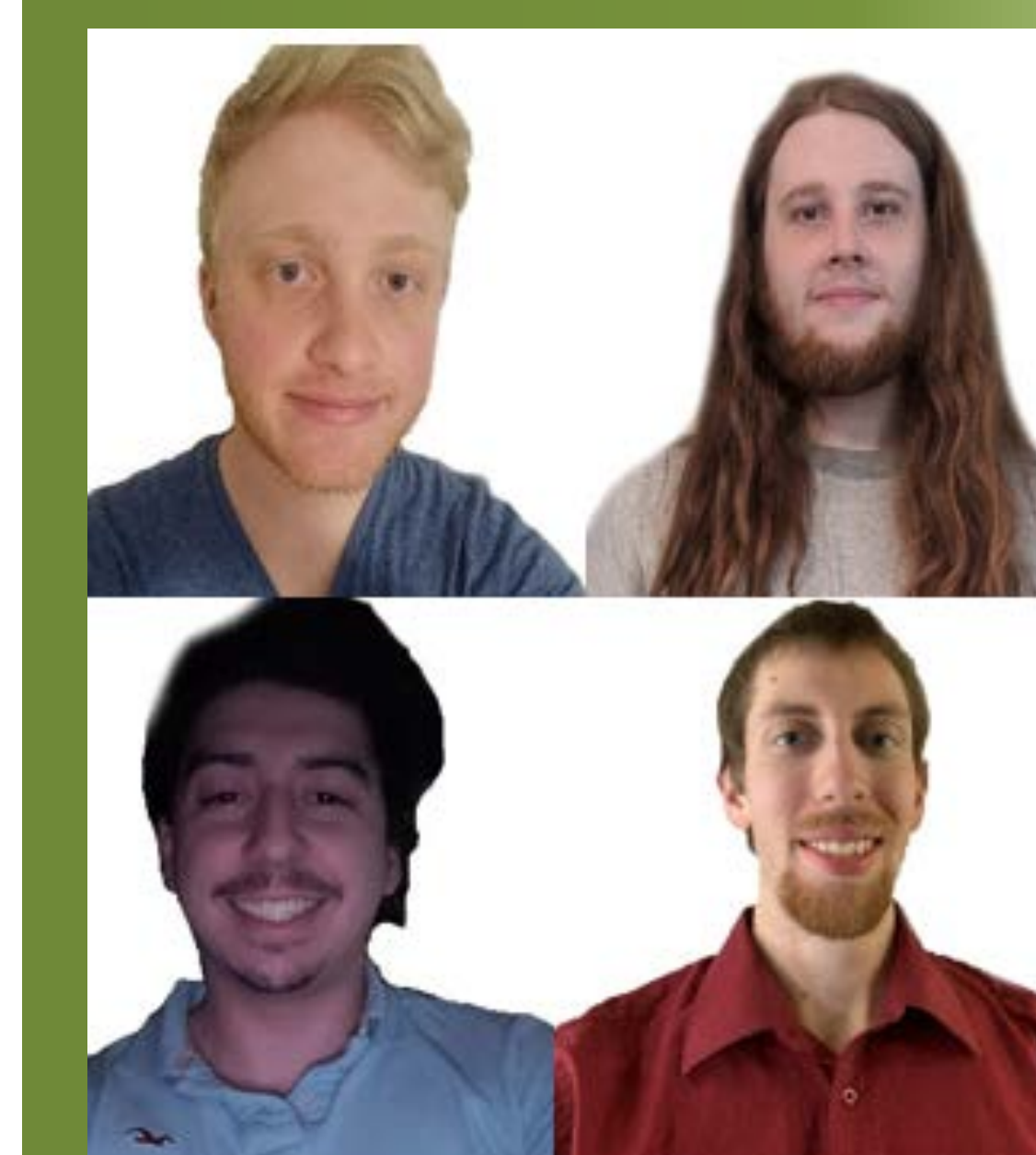
Input File

PostgreSQL Database

Id	Name	Description	↑ Balance
777777	Ben Geiger	7	$7.75
999999	Tom Agnew	CEO of Lease Analytics	$3.50
300789	Jeremiah Spears	Group member of Open Source Generation	$68.27
505554	Billy Bop	He has a lot of moneyy	$1,000,055.00
123456	John Smith	John is a man.	$2.00
654321	Aleck Blohowiak	tall, looks at sky	-$9,000.00

Lessons Learned

We had to familiarize ourselves with new languages and resources we had never used such as Ruby on Rails, Apollo, and Hasura. We also experimented with certain resources such as Webpack, only to find that there were hang-ups in how we were trying to utilize them. The project could be considered a proof of concept, creating a way for web developers to create an infinite arsenal to speed up their dev time. As such, most of our work could be considered a learning experience.

Jeremiah Spears
Ryan Haupers
Aleck Blohowiak
Ben Geiger