In My Element Capstone Project

Shaun Cobb

Jordan Pan

Shane Polwort

Daari Terrell

ChristyAnna Zimmerman

National University

Abstract

A thorough review of available Windows 8 Store applications revealed a potential opportunity for our National University Capstone team (Knowledge Bomb Studios or KBS for short) to develop a comprehensive learning application for chemistry students in the middle through high school range. Aside from completing the Master of Science in Computer Science Capstone requirement of developing a complete working application that fills a niche, this opportunity also allows KBS to establish a market presence and provide an easy-to-use, inexpensive, and intuitive Windows 8 learning application. The KBS team has developed Model-View-ViewModel (MVVM) based software architecture. The Model is programmed in C# while XAML was employed for the View and ViewModel layers, starting from a WPF (Windows Presentation Foundation) template in Visual Studio 2013, and connected to an integrated SQLite database. The final result is an attractive and practical learning application called "In My Element" that allows users to explore and learn about chemical periodic elements at their own pace in Study Mode, challenge their skills in Puzzle Mode, and show mastery in timed Quiz Mode. Quiz Mode further allows users to generate and save their own quizzes for a more focused learning path.

Keywords: C#, Windows 8 Store, SQLite, MVVM, XAML, Chemistry, Periodic Table

Acknowledgements

We would like to extend a heartfelt thanks to all of our National University Professors for their invaluable knowledge and experience and especially to our families for their constant support and understanding of the time and energy required to complete this academic venture. Without the wisdom, guidance, and patience of those mentioned above, this project would not have been possible.

Table of Contents

List of Tables 8
List of Figures 9
List of Acronyms & Definitions 11
1. INTRODUCTION 12
1.1 Motivation 13
1.2 Problem Identification 13
1.3 System Overview 14
1.4 Document Overview 14
2. OVERVIEW OF TECHNOLOGIES EMPLOYED 16
2.1 Architectural Design 16
2.2 Logical Architecture 17
2.3 Physical Architecture 18
3. REQUIREMENTS SPECIFICATION 20
3.1 Functional Requirements 20
3.2 Non-Functional Requirements 23
3.2.1 Availability Requirements 24
3.2.2 Accessibility Requirements 24
3.2.3 Platform Compatibility Requirements 24
3.2.4 Usability Requirements 25
3.3 UI Mock Ups 25
3.3.1 Overview 25
3.3.2 Navigation and GUI Mockups 26
4. UML DIAGRAMS 36
4.1 Use Case Diagrams 36
4.1.1 Overview 36
4.1.3 Use Case Scenarios 38
4.1.3.1 Start Program – UseCase_ID1 38
4.1.3.2 Explore Periodic Table – UseCase_ID2 38

4.1.3.3 User Quiz Question Topic Selection - UseCase_ID3 39
4.1.3.4 Generate Question – UseCase_ID4 40
4.1.3.5 Take Quiz – UseCase_ID5 41
4.1.3.6 Play Puzzle Mode – UseCase_ID6 41
4.2 Class Diagrams 43
4.2.1 Overview 43
4.2.2 List of Classes: The Models 43
4.2.3 List of Classes : The ViewModels 45
4.2.4 UML Class Diagram 47
4.3 Sequence Diagrams 49
4.3.1 Overview 49
4.3.2 Mode Selection Sequence Diagram 49
4.3.3 Add Favorite Elements Sequence Diagram 50
4.3.4 Take Note Sequence Diagram 50
4.3.5 Take Quiz Sequence Diagram 51
4.3.6 Quiz Generation Sequence Diagram 52
4.3.7 Puzzle Mode Sequence Diagram 53
4.4 Activity Diagrams 53
4.4.1 Overview 53
4.4.2 Activity Diagram 54
5. DESIGN AND IMPLEMENTATION 56
5.1 Interface Design 56
5.1.1 Study Mode UI Screenshot 57
5.1.2 Puzzle Mode Screen Shot 59
5.2 Network Architecture 59
5.3 Database Design 60
5.3.1 Database Diagrams 61
5.3.2 Use of Database 63
5.3.2.1 Element Properties Population: 63
5.3.2.2 Quiz Question and Answer Generation: 63

5.3.1.3 User Question and Answer Section: 63
5.3.1.4 Save/Load Quiz: 64
5.3.1.5 User Favorites: 64
5.3.1.6 Note Taking: 64
5.3.2 Table of Definitions 64
5.3.3 Series Table 65
5.3.4 Element Table 65
5.3.5 Property Table 66
5.3.6 Element_Property Table 66
5.3.7 Quiz_Level 67
5.3.9 Quiz_Question 67
5.3.10 User Attributes Table 68
5.3.11 User_Quiz_Question Table 68
5.3.12 User_Note Table 69
5.3.13 User_Favorite Table 70
5.3.14 Saved_Quiz Table 70
5.4 Entity Relationship 71
5.4.1 Overview 71
5.4.2 Entity Relationship Diagram 73
5.5 Database Normalization 74
6. TEST PLAN 75
6.1 Overview 75
6.2 Objective and Scope 75
6.3 Test Deliverables 76
6.4 Exit Criteria 76
6.5 Personnel 77
6.6 Equipment 77
6.7 Testing Schedule 77
6.8 Test Plan Metrics 79
6.8.1 Test Types 79

6.8.2 Program Test Areas 79
6.8.2.1 Start Screen App Launch 80
6.8.2.2 Activity Selection Screen UI 80
6.8.2.3 Study Mode Interactive Periodic Table 80
6.8.2.4 Quiz Mode 80
6.8.2.5 Puzzle Mode 81
6.8.2.6 SQLite Database Access and Manipulation 82
6.8.3 Test Cases 83
6.8.3.1 Start Screen App Launch 83
6.8.3.2 Activity Selection Screen UI 83
6.8.3.3 Study Mode Interactive Periodic Table 84
6.8.3.4 Quiz Mode 85
6.8.3.5 Puzzle Mode 86
6.8.3.6 SQLite Database Access and Manipulation 88
6.8.4 Defect Tracking 90
6.8.5 Test Result Metrics 91
7. USER MANUAL WITH SCREENSHOTS 92
7.1 How to Download the Application and Start the Program 92
7.2 How to Use Study Mode 93
7.3 How to Use Puzzle Mode 95
7.4 How to Use Quiz Mode 97
7.5 How to Generate a User Quiz 99
8. FUTURE WORK AND CONCLUSION 100
8.1 Additional Platforms 100
8.2 Additional Modes 100
8.3 Additional Features 100
8.4 Conclusion 100
9. REFERENCES 102
10. APPENDIX A: SAMPLE CODE 103

List of Tables

Table 1.1 List of Acronyms & Definitions 11
Table 5.1 Definitions 64
Table 5.2 Series 65
Table 5.3 Element 65
Table 5.4 Property 66
Table 5.5 Element Property 66
Table 5.6 Quiz Level 67
Table 5.7 Quiz Question 67
Table 5.8 User Attributes 68
Table 5.9 User Quiz Question 69
Table 5.10 User Note 69
Table 5.11 User Favorite 70
Table 5.12 Saved Quiz 70
Table 6.1 Start Screen Test 83
Table 6.2 Activity Selection Test 83
Table 6.3 Study Mode Interaction Test 84
Table 6.4 Quiz Mode Test 85
Table 6.5 Puzzle Mode Test 86
Table 6.6 Database I/O Test 88
Table 6.7 Defect Tracking Chart 90

List of Figures

Fig. 2.1 Development Technologies Stack 16
Fig. 2.2 MVVM Class Hierarchy 18
Fig. 2.3 Physical Architecture 19
Fig 3.1 Windows 8 Start Menu w/ Favorite Element 26
Fig 3.2 Periodic Table Study/ Information Page 27
Fig 3.3 Notes on Certain Element 28
Fig 3.4 Quiz Mode Selection Screen 29
Fig 3.5 High Scores Page 29
Fig 3.6 Loading a Previously Saved Quiz 30
Fig 3.7 Quiz Generation Screen 31
Fig 3.8 Menu for Advanced Properties 32
Fig 3.9 Menu for Individual Elements 32
Fig 3.10 Screenshot of Quiz in Progress 33
Fig 3.11 Exit Confirmation 34
Fig 3.12 Puzzle Mode 35
Fig 4.1 Use Case Diagram 37
Fig 4.2 Class Diagrams Part 1 44
Fig 4.3 Class Diagrams Part 2 46
Fig 4.4 UML Class Diagram 48
Fig 4.5 Select Mode Sequence Diagram 49
Fig 4.6 Select Favorite Sequence Diagram 50
Fig 4.7 Take Note Sequence Diagram 51
Fig 4.8 Take Quiz Sequence Diagram 52
Fig 4.9 Generate Quiz Sequence Diagram 52
Fig 4.10 Puzzle Mode Sequence Diagram 53
Fig 4.11 Activity Diagram 55
Fig 5.1 Element Selected Study Page 57
Fig 5.2 Element Selected Properties Study Page 58
Fig 5.3 Playing Puzzle Page 59

Fig 5.4 Data Models 61
Fig 5.5 Database Design Diagram 62
Fig 5.6 Database Entity Relationship Diagram 73
Fig 6.1 Testing Schedule 77
Fig 6.2 Test Cases by Phase 78
Fig 6.3 Test Plan Metrics 79
Fig 6.4 Pass/Fail Results 91
Fig 7.1 Activity Selection Screen with Element of the Day 92
Fig 7.2 Activity Selection Screen with Study, Puzzle, Quiz Modes 93
Fig 7.3 Unselected Study Mode Page 94
Fig 7.4 Selected Element's Properties Page 94
Fig 7.5 Puzzle Mode Start 95
Fig 7.6 Populating the Puzzle 96
Fig 7.7 Completed Puzzle 96
Fig 10.1 Element.cs 102
Fig 10.2 QuizQuestion.cs 102
Fig 10.3 StudyElement.xaml 104

List of Acronyms & Definitions

Term	Definition
.NET	Software framework developed by Microsoft.
ERD	Entity Relationship Diagram
Functional Requirements	Defines the inputs, behavior, and outputs of a software system or its components.
IDE	Integrated Development Environment
IME	In My Element; the title of our National University capstone project and associated Windows 8 Store application.
KBS	Knowledge Bomb Studios; our National University capstone team.
GUI	Graphical User Interface
MVVM	Model-View-ViewModel; an architectural pattern targeted at event driven UI development platforms.
RDMS	Relational Database Management System
Role	A combination of the set of permissions and affiliation granted to a user for a title adopted by the user.
SDLC	Software Development Life Cycle; a series of phases that provide a model for the development and lifecycle management of software.
SQL	Structured Query Language; the language of databases.
SQLite	Embedded Database and RDMS integrated into an application.
User	Someone who makes use of an application.
UI	User Interface
UML	Unified Model Language
WPF	Windows Presentation Foundation; graphical subsystem for rendering user interfaces in Windows-based applications.
XAML	Extensible Application Markup Language

Table 1.1

1. INTRODUCTION

Learning can be a daunting task for anyone, especially children and teens, whose attention spans are short, leading quickly to boredom and frustration. Add the analytical complexities of math or the sciences, and what began as an unpleasant and difficult task can become downright arduous and painful. But young people do have age on their side; their attention spans may be short, but the faculties of memorization and recall at these ages are generally the best they will ever be. Modern students also have another distinct advantage: they grew up learning how to use computers and are typically not apprehensive about using them to learn. These characteristics make online learning applications a prime area for the exploration and focus of modern learning techniques.

Our offering to online learning, specifically in the area of chemical periodic elements, is a Windows 8 Store application we lovingly call “In My Element”. Targeted mainly at chemistry students from middle school age through high school, In My Element offers students several available study options, such as an individual examination of each of the chemical elements of the periodic table in a structured form, from a complete periodic table to a study page of a single element. Users can specify favorite elements and can take their own notes about the elements in an available area of the application that provides persistence. In My Element further allows for students to quiz themselves in one or more areas of the complete periodic table: metalloids, non-metals to include halogens and noble gasses, alkali metals, alkaline earth metals, transition and post-transition metals, and lanthanides and actinides. An IME user can have the application quiz them on any one, or even all, of these areas of the periodic table. They can specify a time frame for taking the quiz, and can even save their favorite quizzes for later study.

We chose to develop In My Element as a Windows 8 Store application for several reasons. First, iPhone and Android app stores are already saturated with all kinds of offerings, so we felt Windows provided us with an opportunity to actually get our application in front of students without getting lost in the mire. Second, the entire KBS team are fans of Microsoft's Visual Studio and C# and we wanted to get more exposure to the language. We felt Visual Studio provided a more complete, stable, and well-supported environment than many of the open source Java or web-based development environments we have experienced. As such, we also chose to take advantage of the built-in support for Windows 8 based mobile devices such as smart phones and tablets, but In My Element will be right at home on desktop and laptop computers.

1.1 Motivation

Team KBS' primary motivation was finding a project for our Master's capstone that appealed to each of the team members, provided some utility for potential users, and afforded us all an opportunity to participate while learning.

1.2 Problem Identification

As stated above, the periodic table and the elements can be a difficult subject for students to learn. It is very common for middle school or high school students to have to memorize the elements and their location in the periodic table. Even after memorizing the elements, students may not have an understanding of any properties or uses of the elements. There is a huge amount of information available about the elements, but this can be daunting to students just learning about the elements. It is difficult for students to connect the individual elements to more familiar compounds in the world.

1.3 System Overview

In My Element is an object-oriented C#.NET/WPF based learning application intended for use by Chemistry students from approximately middle school through high school. It includes an element discovery mode to help acquaint students with both the layout of the periodic table and the array of elements that comprise the table. It also includes a quiz mode that can test what users have learned about the elements, as well as a puzzle mode that further tests a user's knowledge of the general arrangement of the table by atomic number. Periodic table data is driven by a backend SQL database that is translated into a XAML based presentation layer using a Model-View-ViewModel (MVVM) architecture.

The In My Element Periodic Table Study Helper is divided into 4 functional areas:

- Frontend GUI presentation
- Backend logic controller
- SQL database
- Integration of frontend GUIs and backend logic

The application platform is a multi-layered architecture to implement presentation, logic, and data layers. The presentation layer implements GUI interfaces to display information to the user. The logic layer implements quiz and puzzle logic and is an intermediary between presentation and data layers. The data access layer is used to store or retrieve information from the database.

1.4 Document Overview

This document is divided into ten sections. Section one provides an introduction and provides insights into our motivation. It also gives a system overview of the project. Section two includes a discussion of the technologies used to develop the IME application. Section three

covers the requirements specification and analysis developed for In My Element and displays the initial UI mockups used for development. Section four includes the UML design diagrams such as class, sequence, and activity diagrams, and also shows the use cases that describe user activities. Section five discusses the design and implementation approach Team KBS used and includes diagrams of the database design for the application. Section six covers the exhaustive testing methods used to ensure In My Element not only met specification and design intent, but covers defect tracking, with unit, regression, integration, and system testing data. Section seven provides a user manual to instruct new users how to navigate through the application. Section eight discusses possible future implementation ideas and provides a conclusion. Section nine lists the references used during project development. Finally, Section ten lists several samples of the code used in the development of In My Element.

2. OVERVIEW OF TECHNOLOGIES EMPLOYED

2.1 Architectural Design

With a multitude of available design patterns for the choosing, the team set out to select one identified as the most appropriate for Windows 8 Store applications running on both desktop/laptop configurations, as well as mobile devices such as tablets and smartphones. As it turns out, that was the MVVM architectural pattern, of which none of the team members had any experience. Based largely on the MVC pattern, the MVVM implementation was designed by Microsoft to target several .NET development paradigms.

As we were planning a user interface application with an event-driven programming model, we opted for a WPF presentation layer written in XAML, with the code behind logic written in C#. See figure below for the software development technologies stack. As these technologies are freely available for use in Microsoft Visual Studio, it was an easy choice to select it as our IDE for this project.

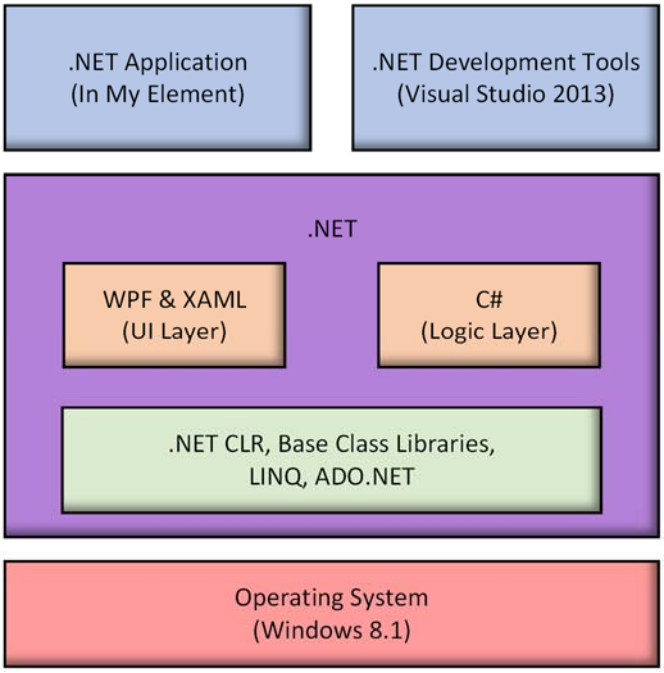

Fig. 2.1 Development Technologies Stack

2.2 Logical Architecture

The In My Element application is a C#/XAML based MVVM platform, which implements a multi-layered architecture as shown in the figure below. The presentation layer implements C# and XAML programming technology using Windows Presentation Foundation forms supported by .NET libraries atop a MVVM architecture pattern, which splits the user interface code into 3 components: Model, View, and ViewModel. The Model includes the classes that represent the data stored in the database. View is the code supporting the visual representation of the data the way it is viewed and interacted with by the user. ViewModel binds the View and the Model together into a cohesive unit. It takes the data from the Model and formats it for presentation by the View. It also controls the View's interactions with the rest of the application.

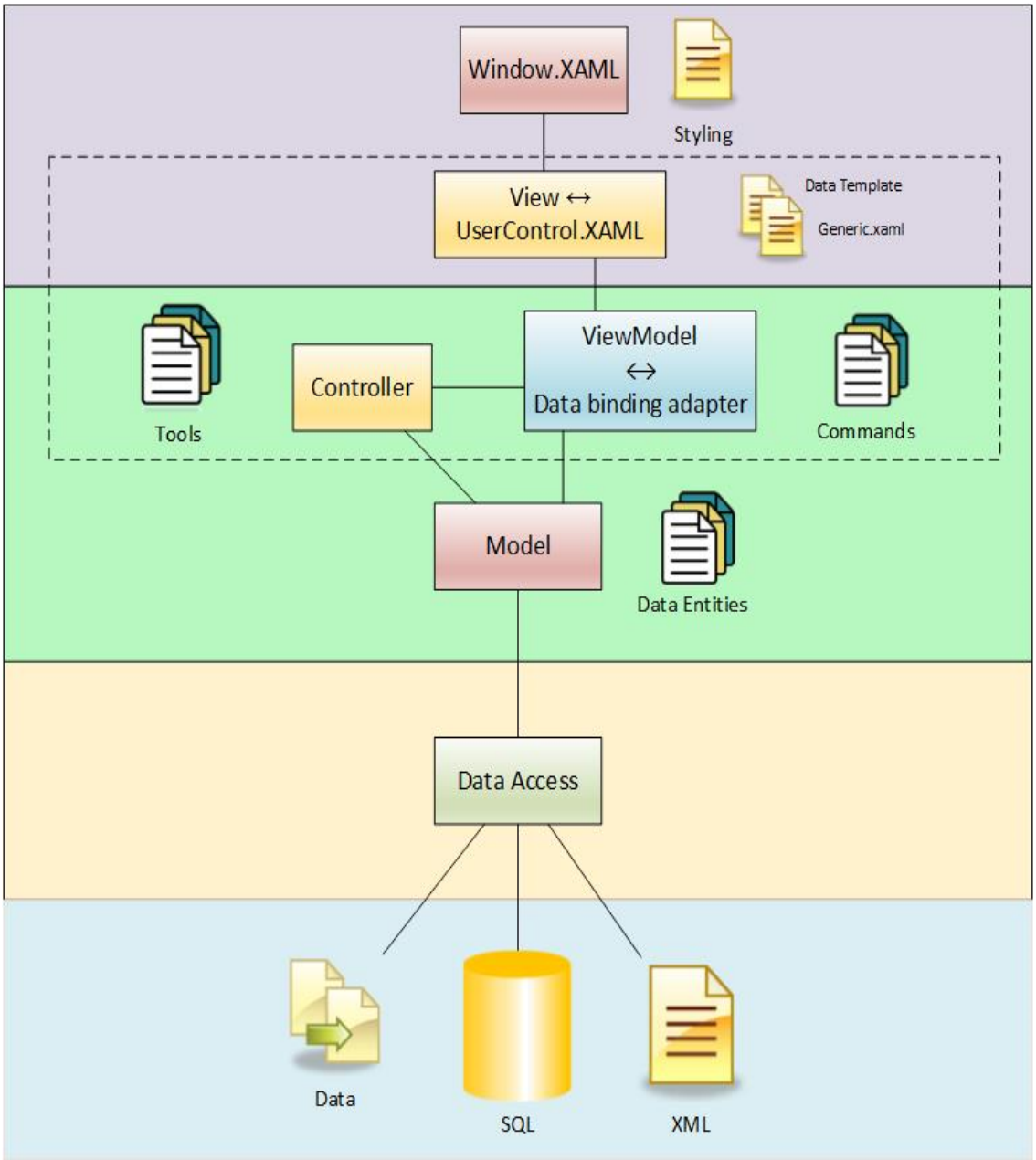

Fig. 2.2 MVVM Class Hierarchy

2.3 Physical Architecture

This application will run on any smartphone, tablet, laptop, or desktop computer running the Windows 8 operating system. It is hosted on the Windows App Store, and can be downloaded for a nominal fee or even for free. We have yet to determine a price, if any. See diagram below.

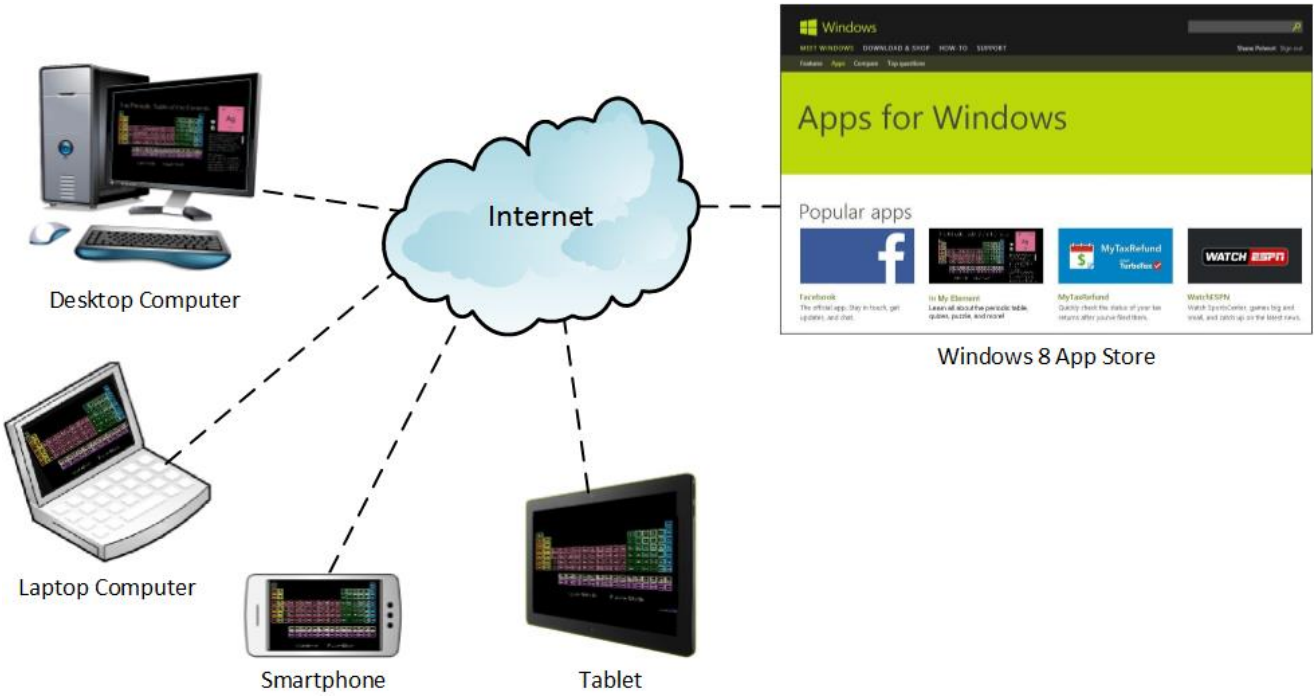

Fig. 2.3 Physical Architecture

3. REQUIREMENTS SPECIFICATION

3.1 Functional Requirements

The In My Element application is an app that is designed for students but can be used by people of all ages. It allows students and working professionals to use it as a reference source, students to use it as a tool to help study for their exams or quizzes, and educators to use it as a tool to help teach their students the periodic elements. This section will go over each of the functional requirements that the application must have. The application will have five sections: a main hub page, a study page, a quiz page, a quiz generation page and a puzzle game page. This section will describe the requirements of the app in a form of free text. UML use case diagrams, among several others, can be found in section 4 of this documentation.

The main hub page will contain a way for the user to navigate to each of the different modes of the application: Study Mode, Puzzle Mode, Quiz Mode and Quiz Generation Mode. The Element of the Day will also be shown on this page, and when user clicks on the link, it will take them to the information for that specific element.

The study page will contain all 118 elements, and each of the elements must be clickable. Once clicked, the information for that element will be displayed either as an overlay, or off to the side. There will also be a button on this page that allows the user to "favorite" one ore more elements. The user must also be able to take notes or make comments about all 118 elements, so there must be an icon or button that gives user access to a form of notepad. From this page, there should be a link to the quiz mode and to the puzzle mode when the user feels the need to test their knowledge of the element through a traditional multiple choice style quiz or through an interactive puzzle game. If none of these icons are present on the page, then the page must

contain a Back button of some sort that allows the user to go back to the hub, where they can choose and where to go next.

The Quiz mode will have two main pages: the actual question and answer pages, and the page where the user can choose which quiz they want to take. In the question and answer page, there must be one question per page that matches the topic the user chooses. There will be 4 choices of answers, and each answer must be clickable. There should be a score board or some type of method for the test taker to keep track of how many questions are left. There will be a timer that can be set, and once the timer runs out, the quiz will end and scores will be calculated. Unanswered questions will be counted as incorrect. A skip question button will be implemented in case the user decides to skip a question. There will also have to be a way for the user to exit the quiz when they choose. Either a back button or an exit button. Once this button is clicked the user should be taken back to the quiz selection page, study page or the main hub page of the app. The part of the quiz mode is the page where the user can choose what type of quiz they want to take. On this screen, there must be a method for the user to choose what type of quiz they want to take based on difficulty or a custom quiz that they have previously generated before. There will be a buttons that allows the users to choose the difficulty settings of the pre-generated quizzes. There will be a button that allows the user to take the "Easy or Beginner" difficulty quiz, a button for the "Medium or Intermediate" difficulty quiz, and a button for the "Hard or Advanced" difficulty quiz. There should be a button where the user can choose a custom quiz that they have generated prior. From this screen there should also be a high scores button that allows the user to check what the high scores are for each of the pre-generated quizzes. When the high score button is selected, it high score page will come up as a pop up window displaying all the scores and best times of the users. Another option to the pre-generated quizzes is allowing the user to take a quiz

that they have previously generated. A button must be present allowing the user to access a list of previously generated quizzes. On this page, the user will be able to delete, edit and start any quiz they choose. The name of the quiz and a brief description of the quiz should also be displayed with each quiz.

Another type of quiz the user can take is a previously generated quiz. To generate that quiz the user must enter the Quiz Generation mode or page. They can enter the quiz generation mode from the quiz mode section with a click of a button. In the quiz generation page, the user will be able to select the number of questions they want to be quizzed on. There will be a text box that allows the user to type in the number of questions and a scroll bar for the user to scroll and select the amount. The user can set the timer for the quizzes in a text box. There will be a scroll bar as well that allows the user to scroll and set the amount of time they would like to have when taking the quiz. The scroll bar will scroll through in increments of 15 seconds and it is displayed in ***mm:ss*** format. There will be a text box underneath that allows the user to disable the timer completely if they choose so. The user also needs to be able to select certain properties of the elements they want to be tested on. Atomic numbers, the name, the symbol, atomic weight, all must be available for the user to select. A "Select All" button must also be present for the user to select all the properties. Also a list must be available for the user to select which elemental group they want to quizzed on. There should also be a method for the user to select individual elements instead of entire groups, in addition there should be a "Select All" checkbox as well. There will also be a text box that gives the user the ability to name the quiz, and another text box for the user to add comments to the quiz. There should be a back button or a cancel button for the incident that the user decides they no longer want to generate a quiz anymore. There should also be a check box, when checked allows the user to take the quiz immediately after saving. A "Save

Quiz" button should be in place for the user to save the quiz when they have checked all the desired categories they want their quiz to contain.

The puzzle mode is what will really separate this app from the others. The puzzle mode provides another tool for the student to learn and practicing the memorization of the periodic table. The puzzle page must have all 118 tiles of the periodic table. The user must be able to click and drag each tile. The tiles are to be arranged in alphabetical order in a list vertically on the side, or placed horizontally at the bottom with a scroll bar for the ones that does not fit on the screen. The atomic number will be placed in each of the "slots" so the user knows what atomic symbol matches with the atomic number. The user must be able to remove the tiles off the board and back into the list if they make a mistake or want to change the placement of a certain tile. A timer will be available to time how long it takes the user to complete the puzzle. The timer only stops when the tiles are in the correct place. When all answers are correct, the date and time will be recorded and be displayed on a high scores page. There will be a reset or clear board button allowing the user to clear the entire board when needed. A "Check Answers" button will be implemented and the validation of the tiles will only be checked when that button is pressed. If the answers are correct, they will turn green, if the answers are incorrect, it will turn red or return itself back to the list. Like all other screens, there must be an "Exit" button in place for the user to quit and return back to the previous menu at any time of their choosing.

3.2 Non-Functional Requirements

Non-functional requirements are requirements that define how a system will perform to carry out their tasks rather than a specific behavior - like a metric to the quality of the system. . Usually these requirements are associated with the system architecture. Non Functional Requirements can be divided into two main categories: Execution qualities and evolution

qualities. Some examples of execution qualities are security and usability and some examples of evolution qualities are testability, maintainability, and scalability which are involved with the structure of the system. The In My Element app is an app that is designed for students and professors of all levels. The app should be always available for users to use, easy for them to obtain the app, accessible to many different devices, and very easy to use.

3.2.1 Availability Requirements

Except for the initial download from the app store, all the features and functions of the In My Element App should be available for the all users 99% of the time. In My Element will be a fully functional app without following the DLC model. The 1% downtime is reserved for the rare case that a new element is discovered, or more information is found and published about certain elements, then a patch will be pushed through to update it.

3.2.2 Accessibility Requirements

In My Element will be submitted to the Windows App Store. Any user that has a Windows 8 desktop, laptop or mobile device and a Microsoft Live account will be able to access the app. There will be a free version that has ads and a .99 cent version that has the ads removed. The app will also feature a minimalist background with very bright clear and concise colors for those with visual impairments. The text will be as large as the display will allow it, meaning the entire program will be scalable according to the device.

3.2.3 Platform Compatibility Requirements

In My Element will only be available for the Windows platform. It will be compatible for desktops, laptops, mobile and tablet users. There are currently no plans to release an iOS and Android version of the application.

3.2.4 Usability Requirements

The In My Element app is an app designed for people with all ages in mind. Whether the user is three years old getting an introduction to the elements, or the user is 80 years young using the app to teach a class or to brush up on the elements themselves, the large intuitive buttons, large easy to read text and minimal graphics will allow the users to navigate and use the app with no instructions required. The In My Element App will feature large easy to use and intuitive navigational menu. All buttons will large, clear and intuitive. It will feature a balance of rich colors and minimalist design approach. Text will be as large as the device allows it, and can be scaled accordingly. With the exception of the Daily Element picture, the app is a straight forward, minimal design following the flat UI design concept.

3.3 UI Mock Ups

3.3.1 Overview

This section will describe the “In My Element” layout and design. Many major companies are designing apps following the flat UI trend. Some notable examples are Google and their layout, Apple and their new IOS design, and the new Windows 8 Metro, Windows 8 RT design. The “In My Element” was designed in such a way that a child could easily navigate their way through the app while following the aesthetics principals of a flat UI design. A flat UI design is a minimalist approach or principal to the layout and design of various graphical interfaces. A flat design contains no shadows and very little to no textures. This is a stark contrast with skeuomorphism, which is the design principal of rendering an object in such a way that it is as close to real life as possible.

3.3.2 Navigation and GUI Mockups

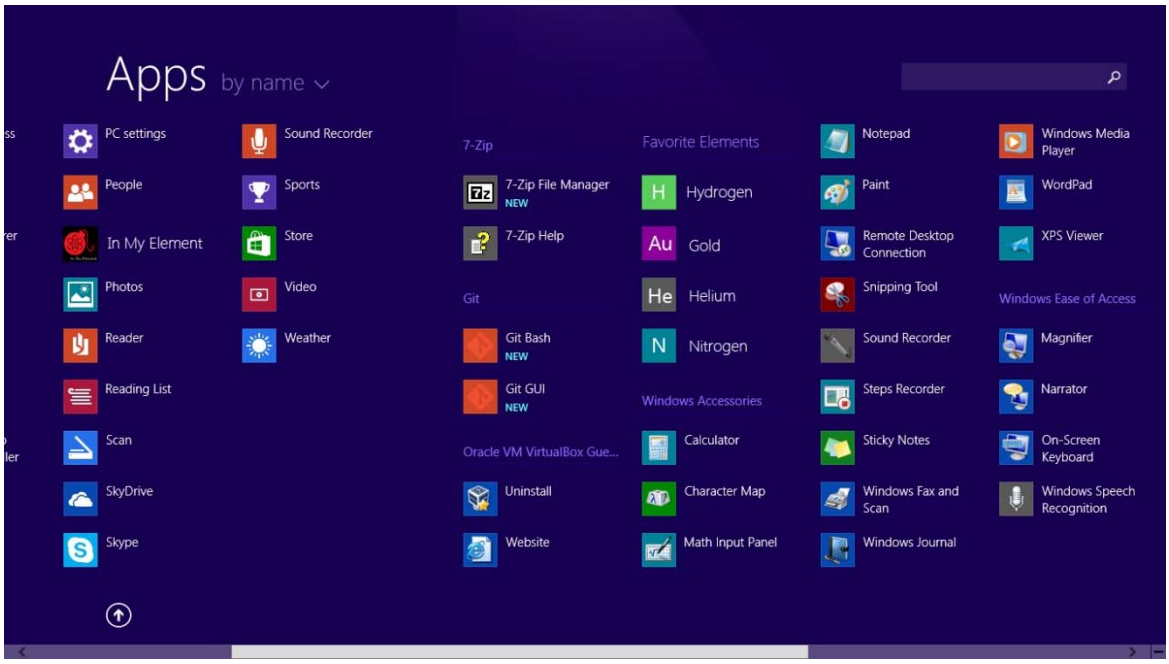

Fig 3.1 Windows 8 Start Menu w/ Favorite Element

The interaction of the app is available without opening the program. The “In My Element” app allows the user to save their favorite elements to the start screen. These favorite elements are pinned to the start menu so they can access the information of the elements when they need to without having to opening up the entire app.

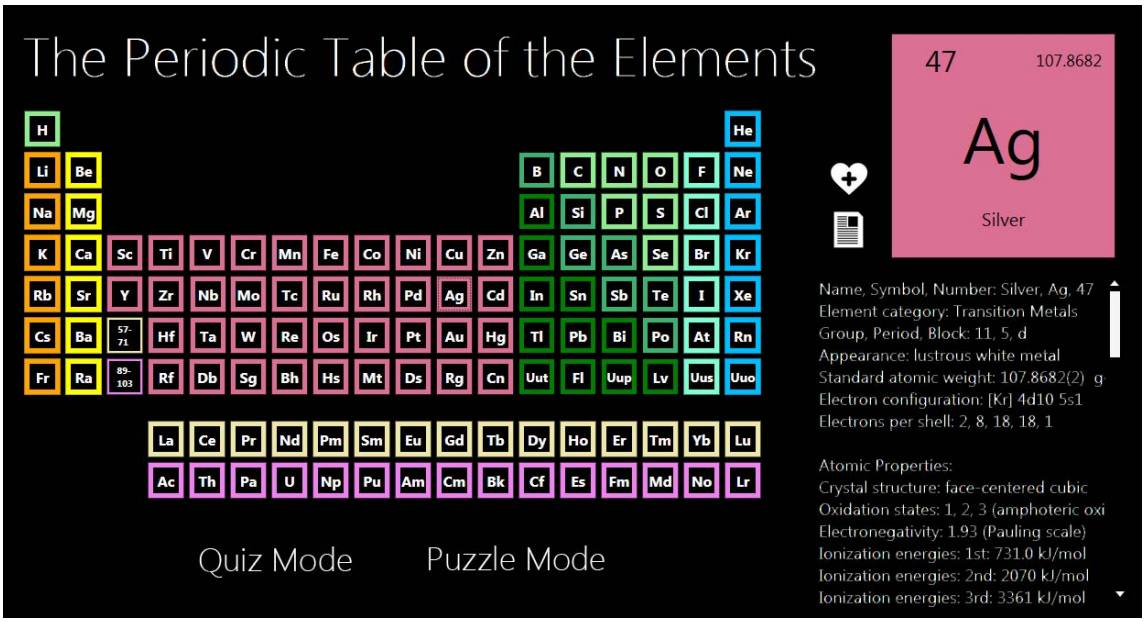

Fig 3.2 Periodic Table Study/ Information Page

Once the user has entered the app, they are presented with the Periodic Table of Elements. Here they are able to click and choose which element they want to read and study on. When an element is clicked the information of the element is presented on the right hand side. Also displayed next to the element is a heart icon, which when clicked allows the user to favorite the element and pin it to the Windows 8 start menu, and it also contains the a document icon, which allows the user to add their own notes or comments to that particular element. Near the bottom of the page will be two navigation button, a “Quiz Mode” and a “Puzzle Mode” button. The “Quiz Mode” advances the user to a different screen where they can take and generate a quiz to help them with their studies. The “Puzzle Mode” button advances the user to a puzzle game, where the user is able to click and drag tiles onto a blank periodic table and test their knowledge and ability through that.

When the user clicks on the document icon, or the “Notes” icon, a text box is presented for the user. With this text box, the user can enter any notes and comments they want about a particular element.

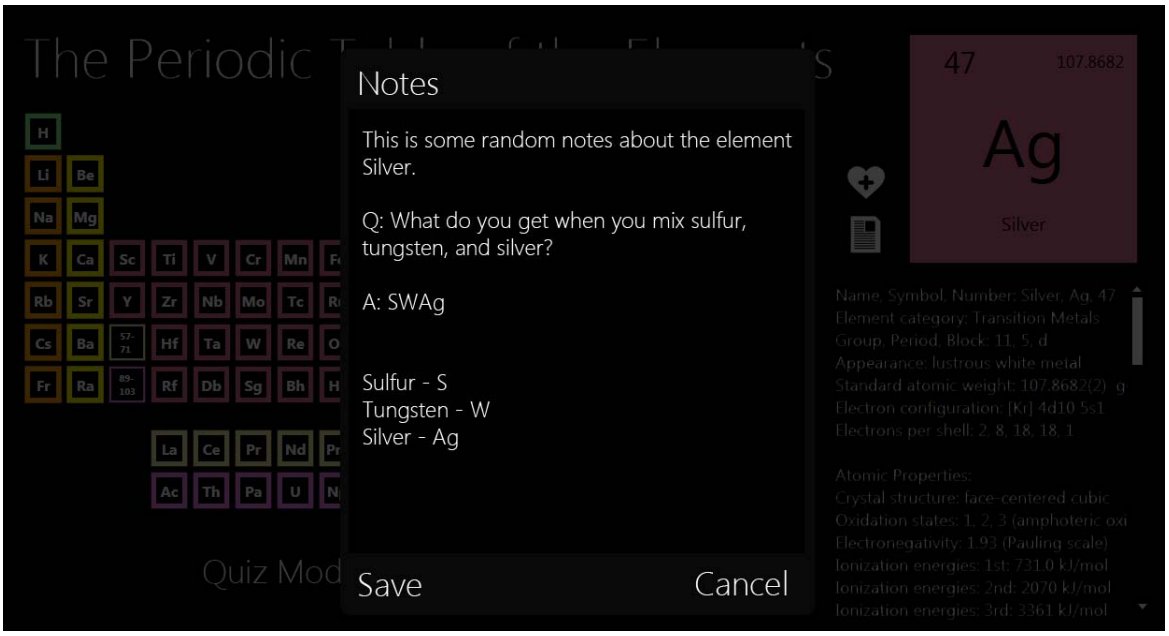

Fig 3.3 Notes on Certain Element

If the “Quiz Mode” button is pressed, the user is then advanced to a screen where they are presented with the options to take a computer generated quiz based on difficulty, load a previously user generated quiz, or create their own user generated quiz. There is also the option of checking the high scores of the computer generated quizzes by pressing the trophy button on the bottom right.

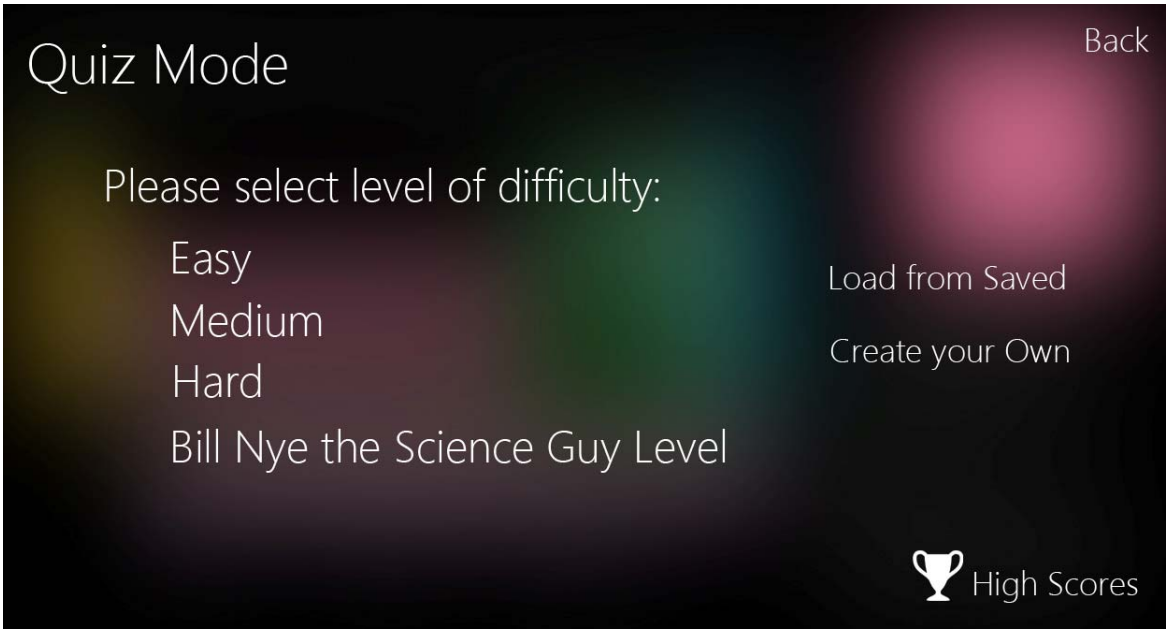

Fig 3.4 Quiz Mode Selection Screen

Fig 3.5 High Scores Page

The High Scores page will show the name of the user who took the quiz, the scores that they achieved and the time that it took for them to finish the test. Only the computer generated

quizzes will have an entry in the high score. The hardest level, “Bill Nye the Science Guy Level” will be displayed significantly larger than the others because of its difficulty.

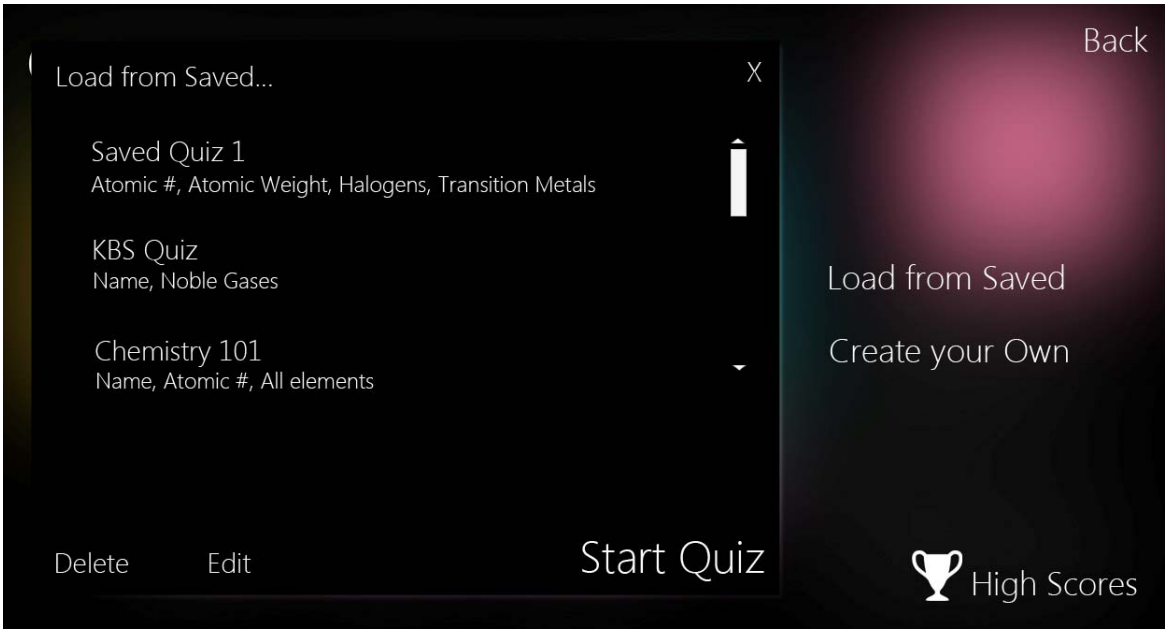

Fig 3.6 Loading a Previously Saved Quiz

At the “Load from Saved” Screen, the user is able to scroll down a list of user generated quizzes. From there, they have the option to delete, edit or take the quiz that they have selected. The Quiz Generation Screen is accessed by the “Create your Own” button found in the Quiz Mode of the app. It is divided into three parts, the left, middle and the right.

On the left side of the screen, a scroll bar will be present allowing the user to select the number of questions they want on their quiz, or they can manually type the questions in. There is also the option to set a timer for the quiz with a check box to disable the timer is the user feels like they do not want to be timed on their quiz. The user will also have the ability to customize what topics they would like to be quizzed on. A list of check boxes will be present where they could select and be quizzed on such things as Atomic Weight and Name of the elements. More advance properties such as “Crystal Structure” is available for the user as well and can be

accessed through the “Advanced Properties” button. A drop down menu will be presented when the button is pressed and from there the user can select from the many advanced topics they want to be tested on.

The middle section allows the user to select which elements they want to be tested on specifically. It is broken down by groups, and they have the ability to select all the elements to be tested on. If the user requires a higher level of detail for their quiz, they can selected “Select Individual Elements” at the bottom of the list and from there a screen will pop up allowing the users to choose which element they want to be tested on specifically.

The far right section of this app allows the user to name their quiz and add comments to the quiz. After creating their quiz, there is a check box present, allowing the user to take the quiz immediately after creation if they choose so.

Fig 3.7 Quiz Generation Screen

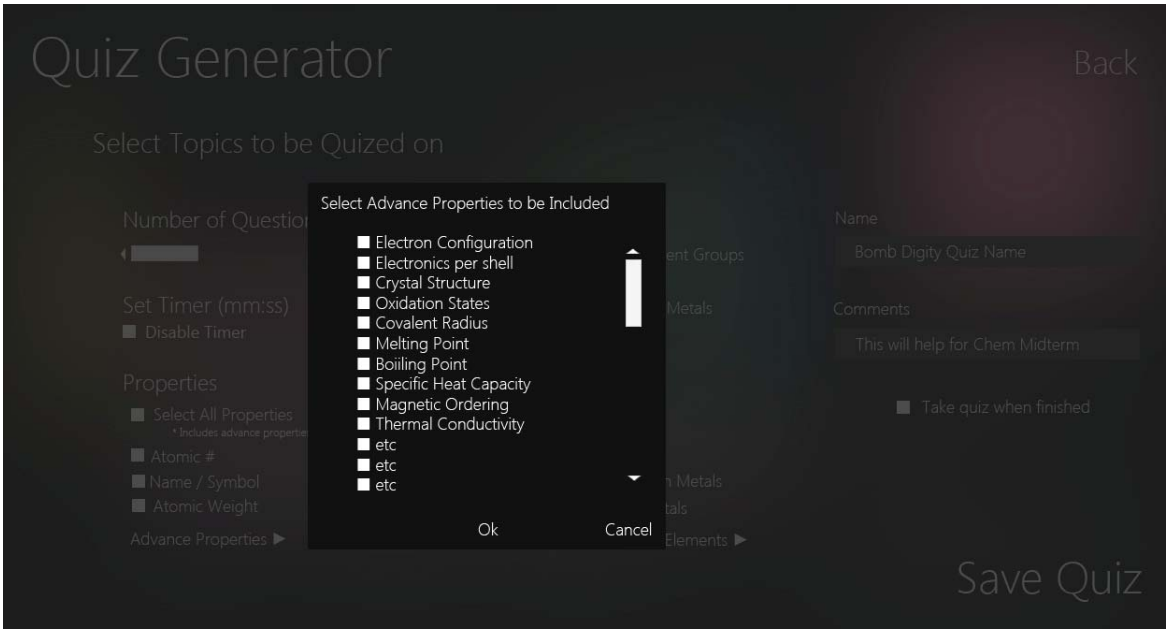

Fig 3.8 Menu for Advanced Properties

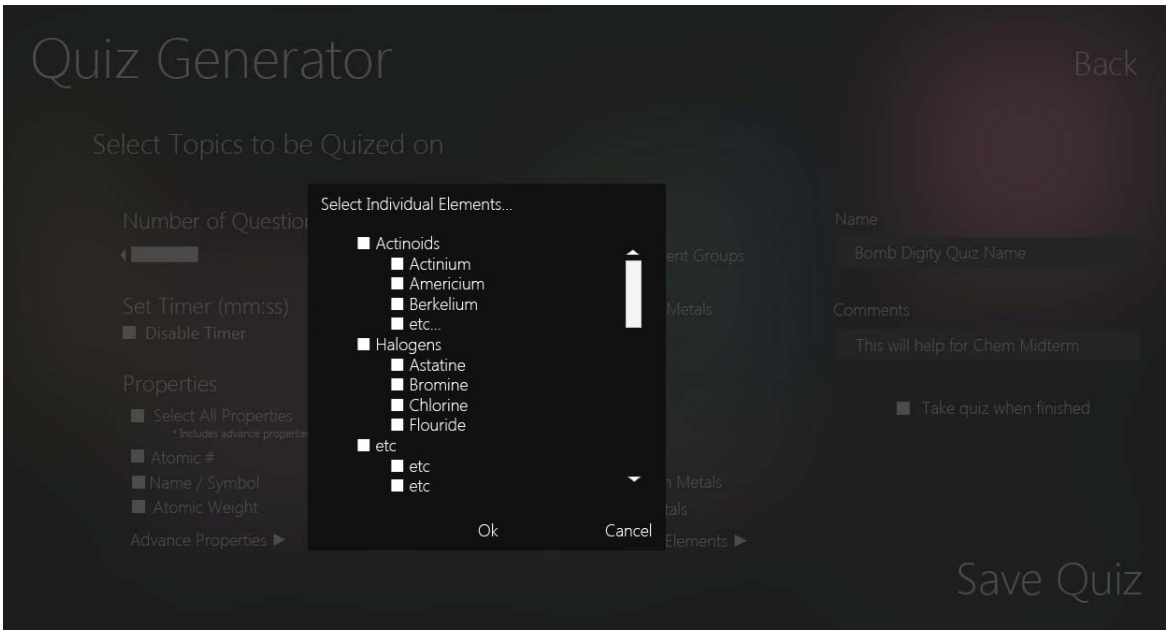

Fig 3.9 Menu for Individual Elements

During the quiz, the user will be presented with the problem and four possible solutions to the problem. The upper right hand corner will display what question the user is on, and how many questions they have left. In addition, it will also display the amount of time the user has left. Nothing will be displayed if the user decided to disable the timer. The bottom left corner allows the user to skip a certain question or exit the quiz entirely. When the “Exit Quiz” button is selected, the user will be prompted with a screen asking them if they are sure they want to exit.

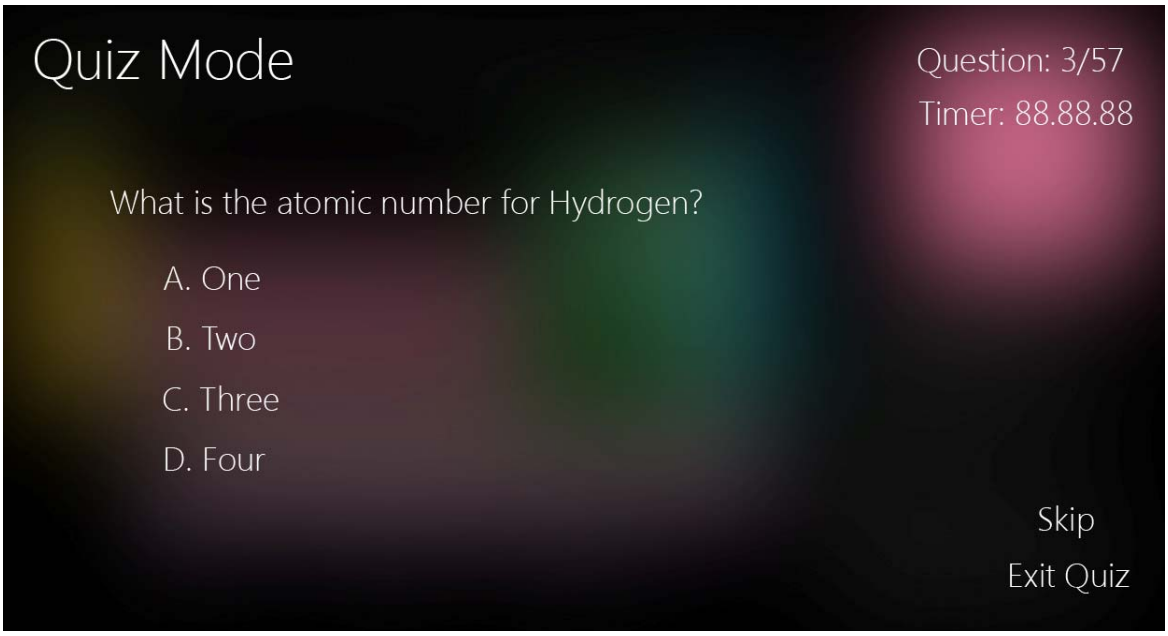

Fig 3.10 Screenshot of Quiz in Progress

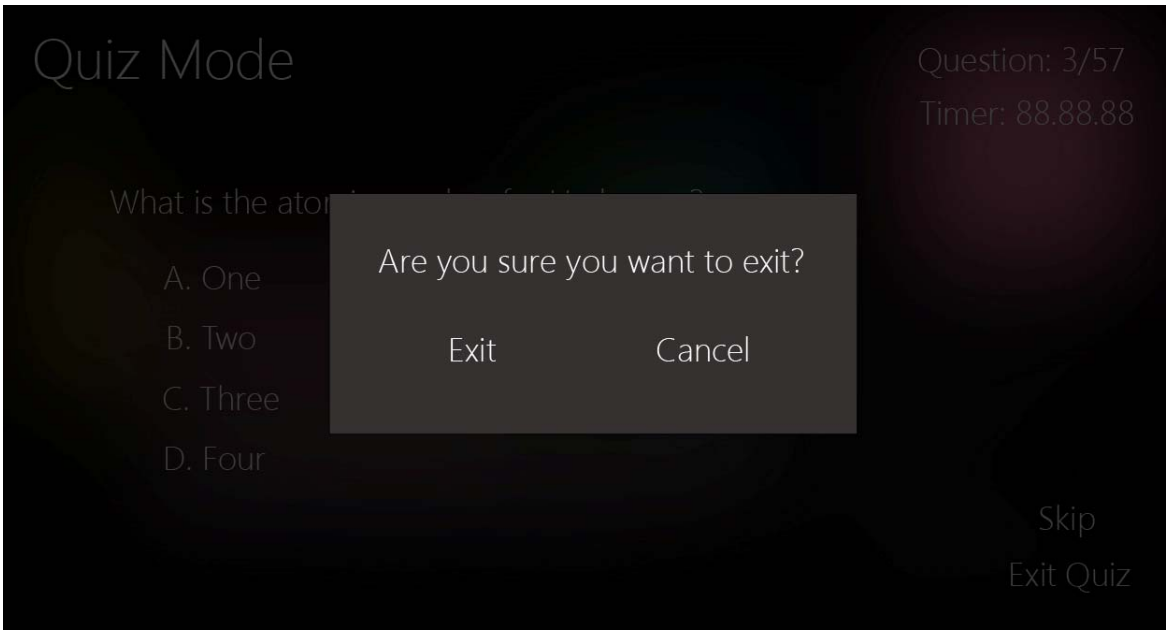

Fig 3.11 Exit Confirmation

The interactive puzzle mode screen allows the user to select and drag the tiles to a blank periodic table in attempts to fill in the blanks. The period table will have a square outline with all the squares representing each element. In each of those squares, the atomic number will be written allowing the user to know what tile should go where. The right hand side of the screen is where all the tiles that are not in play are stored. They are arranged alphabetically by their atomic symbol. Once the tiles are in place, the user can check their answers but pressing the “Check Answers” button located in the bottom center. If the answer is correct, then the tile will light up according to their elemental group with black text. If the tile in at an incorrect spot, the tile will light up a bright red with white text. Once all the tiles are in the correct place, and the check answers button is pressed, the entire periodic table will light up with their respective colors with black text. If at any point the user wants to reset the board, a “Reset” button is located at the

bottom left of the screen. Also, if at any point, the user wants to exit the game and return back to the Study Page, the “Exit” button is located at the upper right.

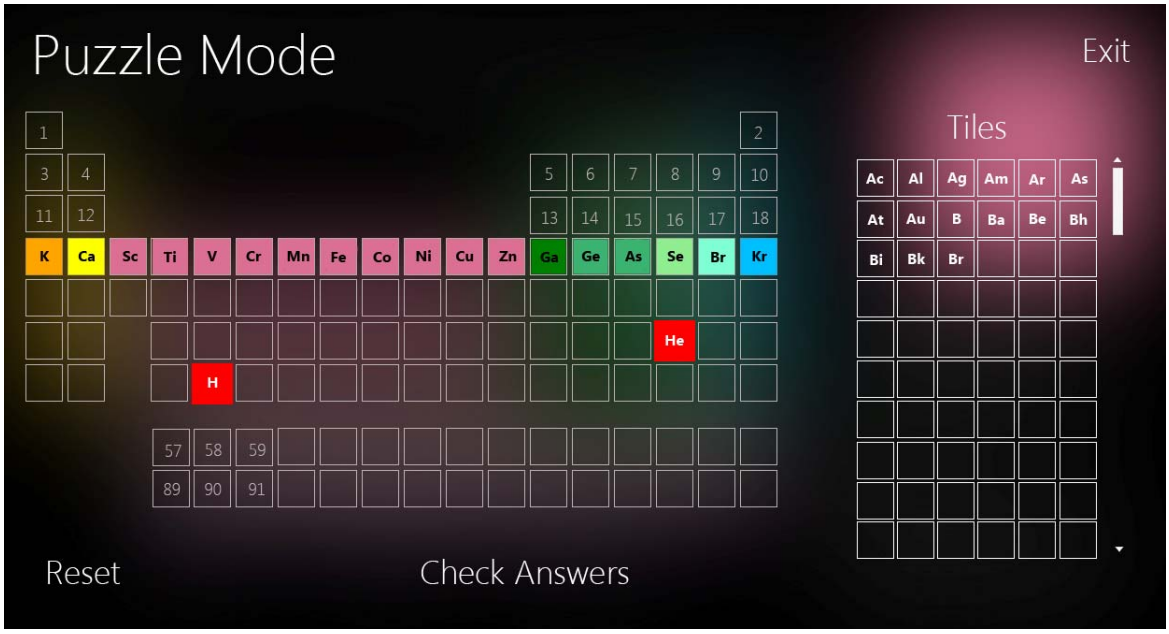

Fig 3.12 Puzzle Mode

4. UML DIAGRAMS

4.1 Use Case Diagrams

4.1.1 Overview

In My Element is an app designed for students and professionals of all ages who want to learn, study and practice the elements found in the periodic table. The use case diagrams are a behavioral UML diagram that represents the user's interaction with the system and various actors relating to the system. Use Case diagrams are usually done in the early phases of development, during the requirements gathering phase because it allows the developer to understand how the program will be used ensuring all functions and features will be developed. This section will display the use cases diagrams and textual use cases displaying the functional requirements of the In My Element App.

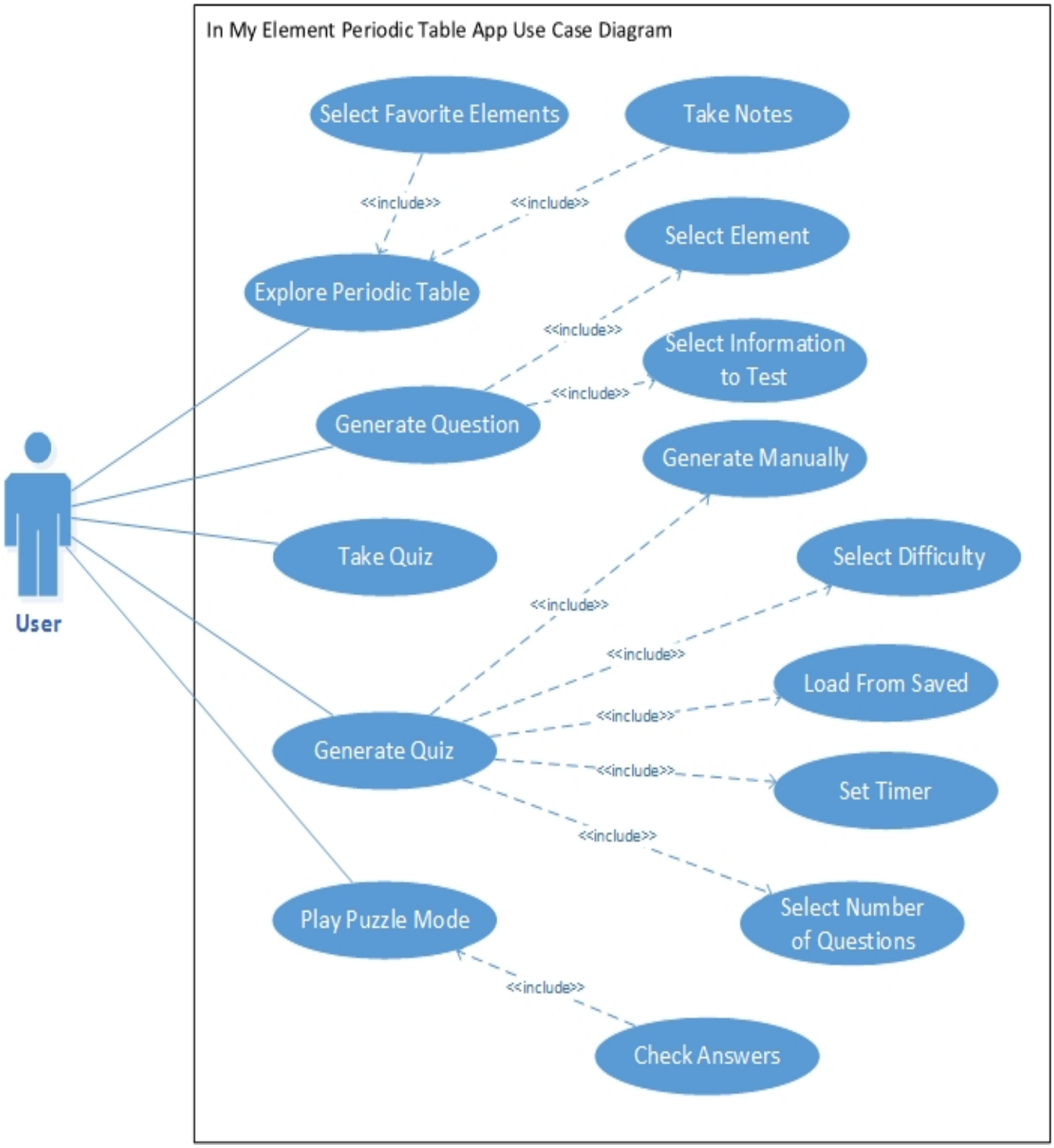

Fig 4.1 Use Case Diagram

4.1.3 Use Case Scenarios

4.1.3.1 Start Program – UseCase_ID1

Use Case Name: Start Program – UseCase_1

Participating Actors: User

Description: User clicks In My Element application tile to start the application\

Precondition: User is at start screen.

Entry Condition: User must click the application tile in order to enter.

Exit Condition: User has successfully transitioned to the activity selection screen.

Quality Requirement: Splash screen for the application should be viewed, along with a loading icon.

Primary Flow of Events:

1. User Selects In My Element Tile from Start Screen.
2. Application transitions to In My Element Splash Screen with a status showing that it is loading the application.
3. Screen transitions to Activity Selection screen along with showing the element of the day.

Alternate Flow 3A: Exit

1. User clicks the X button at the top right of the application app bar to exit the program.

4.1.3.2 Explore Periodic Table – UseCase_ID2

Use Case Name: Explore Periodic Table – UseCase_ID1

Participating Actors: User

Description: User is given the ability to study each of the elements on the periodic table and they are also able to choose their own favorite elements and add notes/comments to individual elements

Precondition: Application has been started.

Entry Condition: User must click the picture representing the activity in order to enter.

Exit Condition: User is presented with a new screen showing the periodic table with a random element highlighted and its information displayed.

Quality Requirement: The random element chosen should indeed be random, and not show the same element consecutively. All should occur as soon as the activity is opened.

Primary Flow of Event:

1. User selects Interactive Periodic Table from main activity screen.
2. Application proceeds to the open the activity, and presents user with a fully rendered periodic table.
3. Application randomly selects an element on the periodic table and presents User with information about the element.
4. User begins selecting the element of their choosing.
5. Element will be highlighted and its information will be shown in a field.
6. User reads information.
7. User can mark element as a favorite
8. User can add notes/comments to the element
9. Repeat Step 4.

Alternative Flow 6A1**: Exit**

1. User exits activity and returns to activity selection page or exits application altogether.

4.1.3.3 User Quiz Question Topic Selection - UseCase_ID3

Use Case Name: User Quiz Question Topic Selection – UseCase_ID3

Participating Actors: User

Description: User has the ability to create their own quiz from the predefined questions

Precondition: Application has been started and user in Quiz generator.

Entry Condition: User must click the on the Quiz button.

Exit Condition: User has made selection, and is ready to take Quiz or Exit.

Quality Requirement: The random question chosen should indeed be random, and not show the same question consecutively nor twice in a session. Secondly, questions must reflect topics chosen by the user.

Primary Flow of Event:

1. User is prompted to select difficulty for the quiz and makes selection.
2. User is prompted to select the number of questions that they want to be tested on chooses a value.
3. User selects the amount of time needed for the quiz.

4. User is prompted to select the element types, element properties and element groups they would like to be quizzed on (Noble Gas, Alkali/Alkali Earth etc.), and selects the element types.
5. User will select to Save the Quiz.
6. User makes selection to generate quiz.

Alternative Flow 2A1: Load Quiz

1. User loads from a pre-saved quiz from before

Alternative Flow 3A1: Enters 0

1. User inputs 0, for no timer or infinite time
2. Go to step 4.

Alternative Flow Any Step: Exit

User exits activity and returns to activity selection page or exits application altogether

4.1.3.4 Generate Question – UseCase_ID4

Use Case Name: Generate Question - UseCase_ID4

Participating Actors: User

Description: User develops their own question based on a topic and an element series that the User chooses.

Precondition: Application has been started. User selected Quiz Mode Activity.

Entry Condition: User selects the Quiz Question Generator option from the Quiz Mode main menu.

Exit Condition: User has created a question.

Quality Requirement:

Primary Flow of Event:

1. User selects a topic from a drop down menu and an element series from a drop down menu.
2. User types in question, and its corresponding set of 4 answers with 1 being labeled as correct.
3. User clicks "Generate Question" button.
4. User selects "Generate another Question".

Alternative Flow Step 4A1: Quiz Mode Main Menu

1. User hits back button to return to main menu.

Alternative Flow Any Step: Exit

User exits activity and returns to activity selection page or exits application altogether.

4.1.3.5 Take Quiz – UseCase_ID5

Use Case Name: Take Quiz– UseCase_ID5

Participating Actors: User

Description: User selects answers to computer generated questions

Precondition: Quiz Activity has been started.

Entry Condition: User has generated a quiz from the Quiz Generation Screen or loaded a quiz from the Quiz Mode Main Menu.

Exit Condition: User has answered all of the generated questions and received a score.

Quality Requirement: The random question chosen should indeed be random, and not show the same question consecutively nor twice in a session.

Primary Flow of Event:

1. User is present with a screen showing the first question at random and its corresponding set of answers
2. User then selects an answer, and presses a button that allows them to proceed to the next question.
3. Application tallies score and presents User with another question and another set of answers.
4. Repeat Step 3, until final question is reached.
5. Application shows final score and grade.

Alternative Flow 2A1: Skip

1. User chooses skip button.
2. Return to Step 3.

Alternative Flow Any Step: Exit

User exits activity and returns to activity selection page or exits application altogether

4.1.3.6 Play Puzzle Mode – UseCase_ID6

Use Case Name: Play Puzzle Mode– UseCase_ID6

Participating Actors: User

Description: User selects Puzzle activity

Precondition: Application has been started.

Entry Condition: User must click the picture representing the activity in order to enter.

Exit Condition: User is presented with a new screen showing a blank periodic table and a group of tiles to choose from.

Quality Requirement: Table must be completely blank. All tiles must be accessible. All tiles must be drag enabled and droppable.

Primary Flow of Event

1. User selects Puzzle from activity selection screen.
2. Application proceeds to the open the activity, and presents user with a completely blank table and a complete list of all the element tiles in alphabetical order.
3. User then selects a tile, and drags the tile to its appropriate box.
4. Repeat step 3 until final tile is reached.
5. When final tile is placed, User selects a button that will check their answers.
6. When table is complete, Application will show User the time to completion for the current session.

Alternative Flow Any Step: Exit

1. User exits activity and returns to activity selection page or exits application altogether.

Alternative Flow 3A1: Reset

1. User chooses the reset button.
2. Go to step two

Alternative Flow 5A1: Wrong Answers

1. Application returns to the list some of the tiles that were placed in the wrong boxes.
2. Go to step 3.

Alternative Flow 6A1: Reset

1. After the Application gives a time score and comparison of times, User chooses the reset button.
2. Go to step 2.

4.2 Class Diagrams

4.2.1 Overview

A Class Diagram is the visual representation of the static structure of a particular application or system. It describes the structure of the system by illustrating the system's classes, including their attributes, operations, functions, methods and the relationship these classes have with other classes or objects in the system. The default design pattern that all Windows 8 applications follow is the Model View ViewModel design pattern. The Model View ViewModel pattern allows the developer to easily separate the application logic, objects and data (the Model) from the user interface (the View). The ViewModel is the controller that contains the commands, functions and logic necessary to encapsulate both the user interface and the data and objects of the software. The class diagrams presented in this section displays the structure of the various classes and objects created for the In My Element app. It displays how these classes and objects relate to each other while following the Model View ViewModel design paradigm.

4.2.2 List of Classes: The Models

Shown below are a list of some of the classes created in for the In My Element application. One of the most important class in the application is the "Element" class. In My Element being a periodic table application, relies heavily on the "Element" class as all the information about each particular element is contained in this class. This diagram shows the content of the individual classes only. A relational class diagram can be found below.

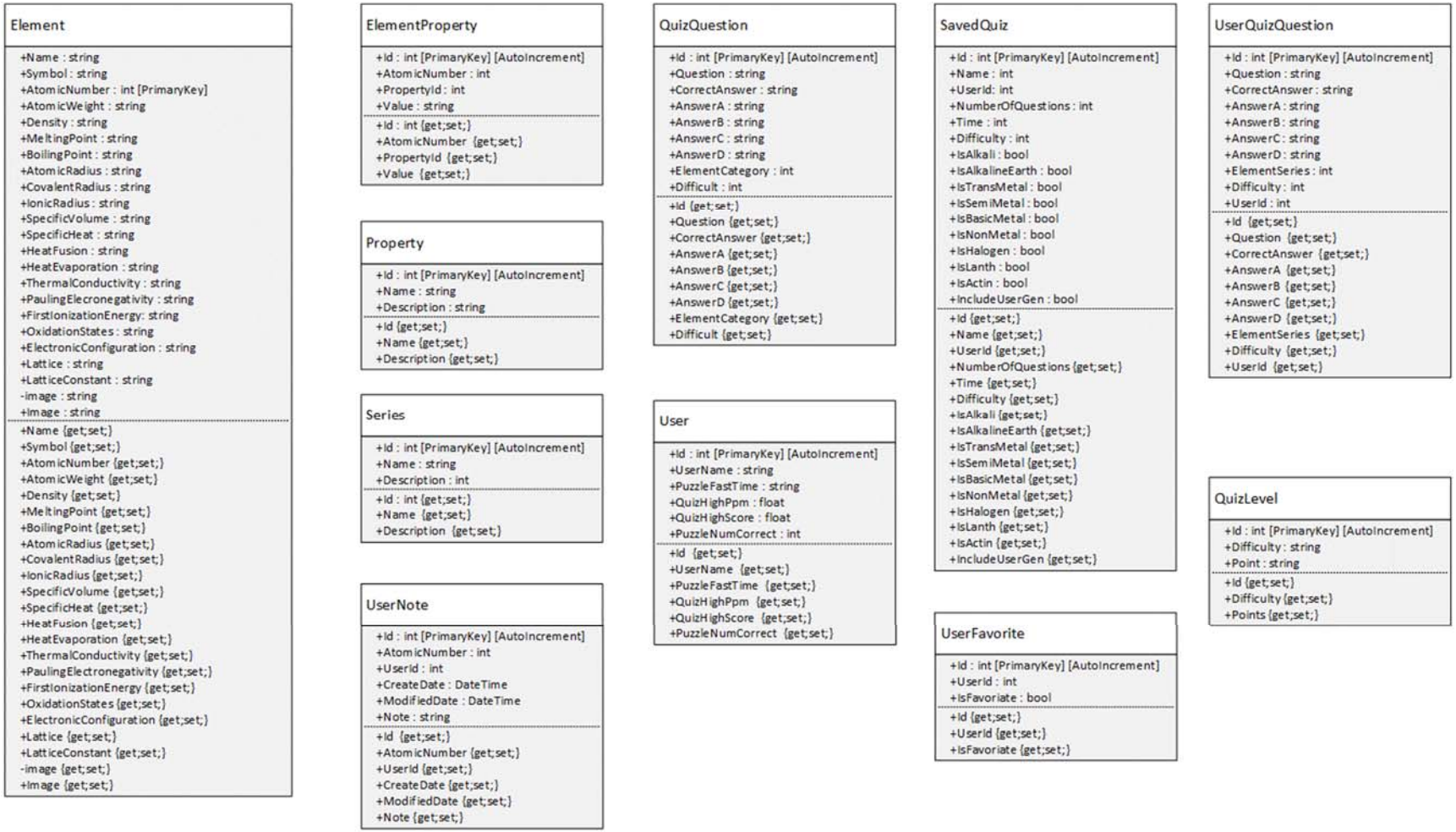

Fig 4.2 Class Diagrams Part 1

4.2.3 List of Classes : The ViewModels

This diagram contains a list of the View Models of the In My Element application. A View Model is similar to the controller in a MVC (Model-View-Controller) paradigm. The "IDataRepository" class is the class that helps access the data repository of the quizzes and the element. The "SQLiteRepository" class helps with the access of the SQLite database that is implemented. They both contain the methods and attributes needed to perform the various tasks presented to them. "PuzzleViewModel" class is the View Model responsible for the Puzzle mode of the application. It provides the link between the Puzzle View and the Puzzle Model classes. The "ViewModel" class is the View Model that is responsible for the rest of the application. It provides the link between the various Models and Views present in the In My Element application.

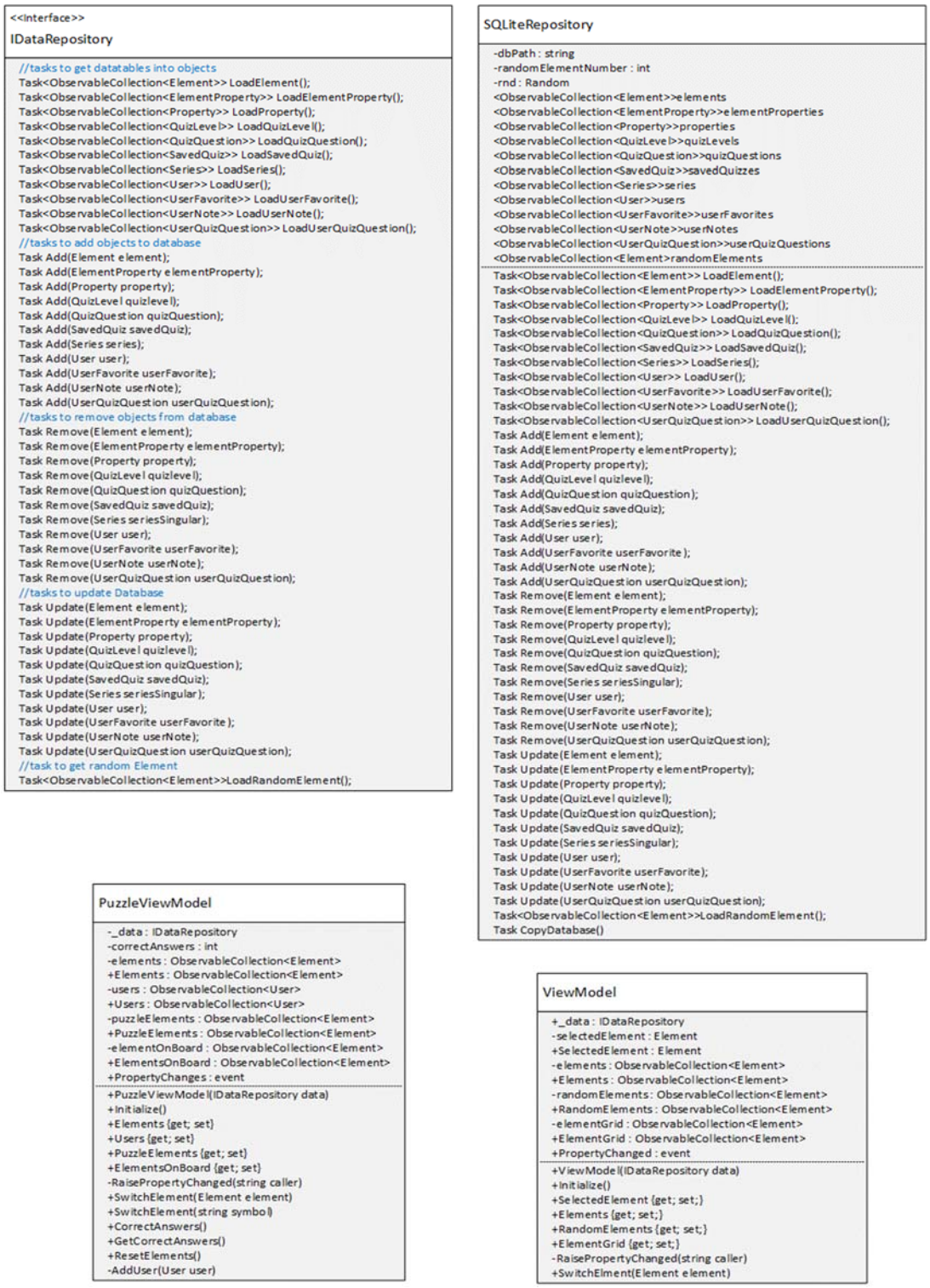

Fig 4.3 Class Diagrams Part 2

4.2.4 UML Class Diagram

A relational class diagram shows the static relationships various classes have with each other. As mentioned before, the "Element" class is an important class in the In My Element application. Many other classes draw their information from that class by association. Due to the size constraint of an A4 sized paper and keeping with the American Psychological Association (APA) styling citation and format, the class diagrams presented below does not contain the details of its methods and attributes. Details of each individual classes will be found in Figures 4.2 and Figures 4.3. Figure 4.4 below just shows the visual representation of the static structure and composition of the In My Element App.

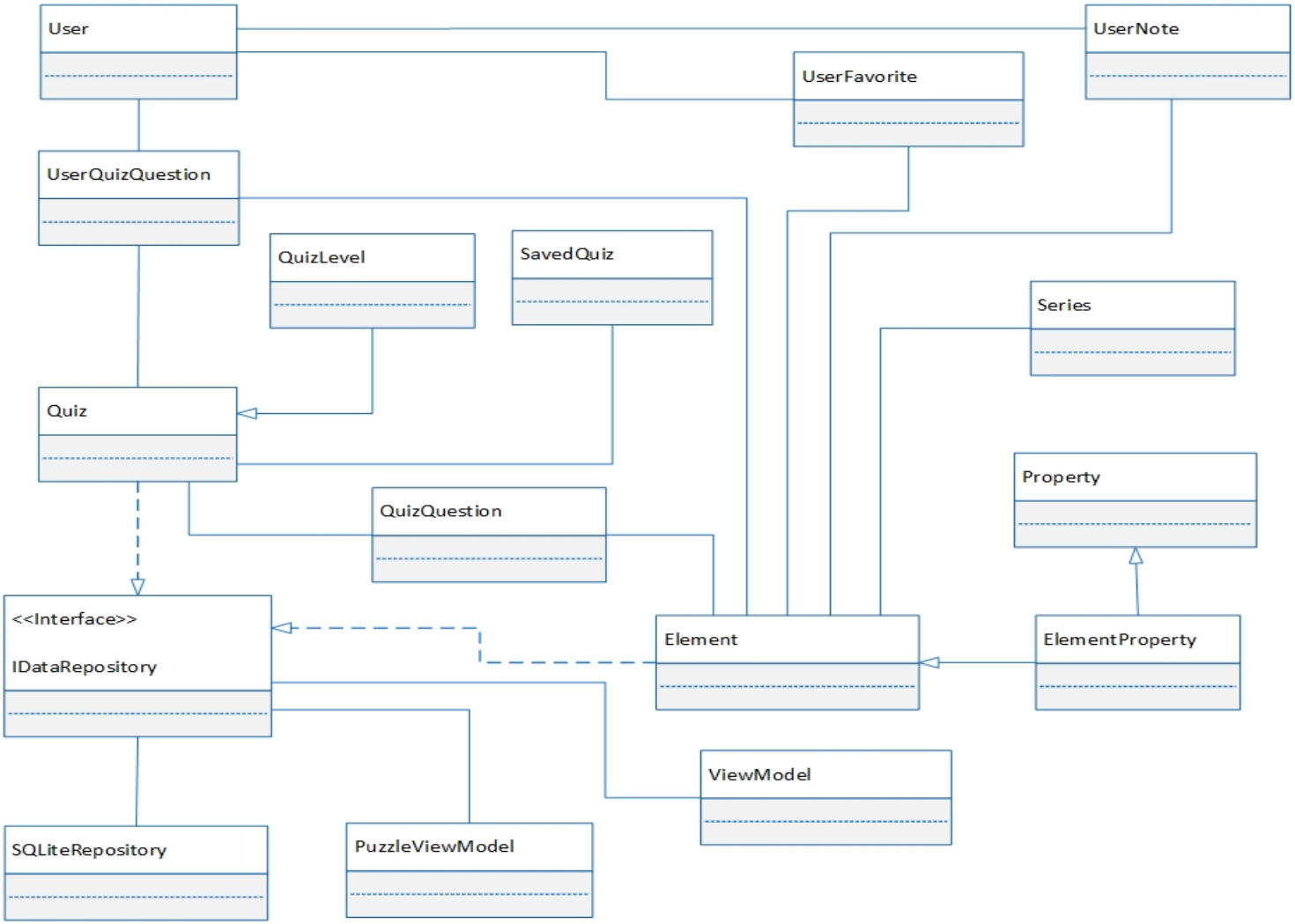

Fig 4.4 UML Class Diagram

4.3 Sequence Diagrams

4.3.1 Overview

Sequence Diagrams shows the interaction between a class, or object, to another in a sequential order. This is not the same as a class diagram above which shows the relationship between the classes and objects. Sequence diagrams are usually closely associated with the use cases of the particular system, meaning there is a sequence diagram for each use case of the system. The sequence diagrams show the classes involved in particular scenario or use case. It shows the actions each class performs, what messages or data involved with the exchange and exactly how it is carried out through the process. The following diagrams show the various interactions the classes and objects have in the In My Element application.

4.3.2 Mode Selection Sequence Diagram

The Mode Selection sequence diagram depicts the shows the interactions involved to select a certain mode from the hub page. The user selects a certain mode from the View. The View then alerts the ViewModel that the user performed an UI action. The ViewModel then fetches the necessary data needed to load the next screen from the Model and passes the information back to the View to be displayed.

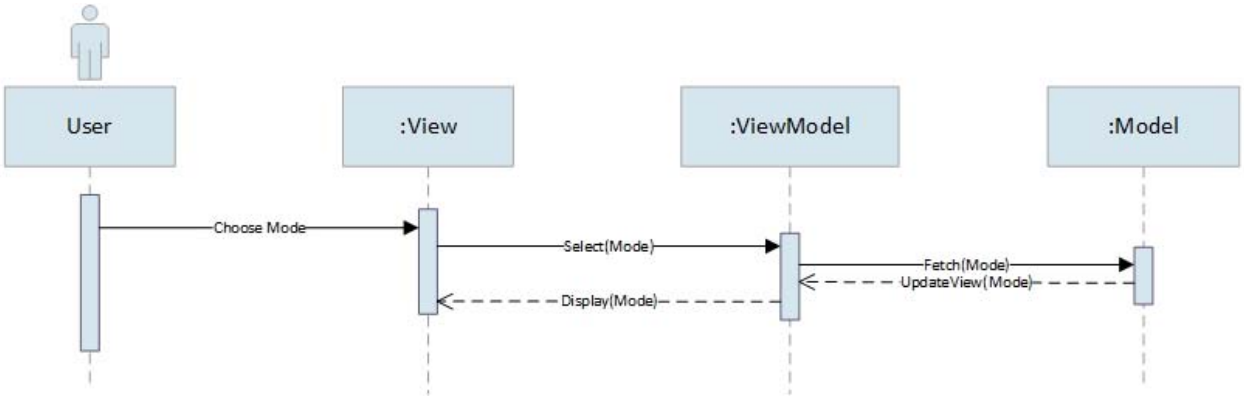

Fig 4.5 Mode Select Sequence Diagram

4.3.3 Add Favorite Elements Sequence Diagram

One of the great features of the In My Element application is the ability for the user to create favorites of their favorite elements. These elements will be pinned to the desktop so they have easy access to it. Once the user selects the Favorites icon in the Study Page, the View Model gets alerted by the View that the user has clicked it. The ViewModel then passes the flag to the Model, which then informs the database and sets the "IsFavorite" boolean logic to true. Then it turns and the element is favorite to the desktop.

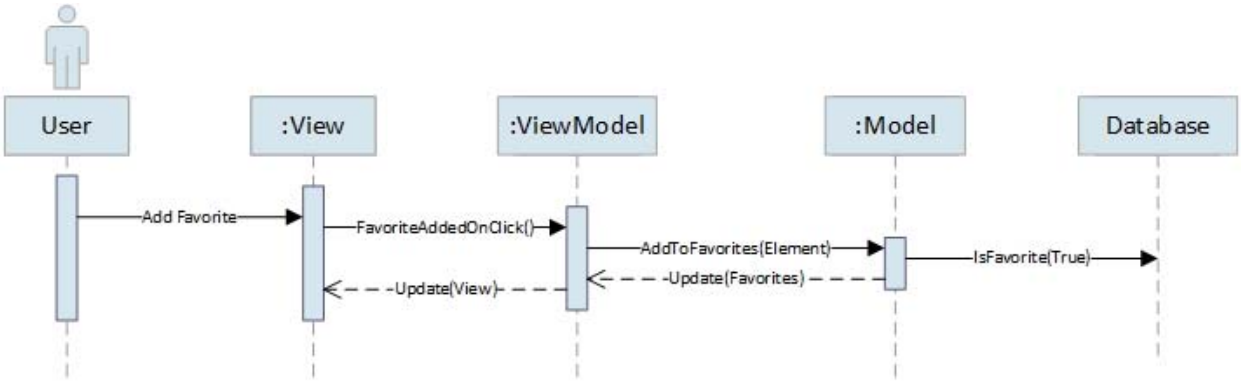

Fig 4.6 Select Favorite Sequence Diagram

4.3.4 Take Note Sequence Diagram

Another feature added to Study Mode is the addition of user notes. Rather than writing notes outside of the application, users can add an electronic note and have it persisted in the database. The note taking sequence diagram follows a similar sequence. When the user clicks the notepad icon, a text box will appear allowing the user to type out their notes at the particular element. When the user hits "Submit" button, the ViewModel will then be alerted by the View that the button has been clicked and the data will be passing through. The data gets passed through to the Model, and the Model then takes the data and updates the Database with it.

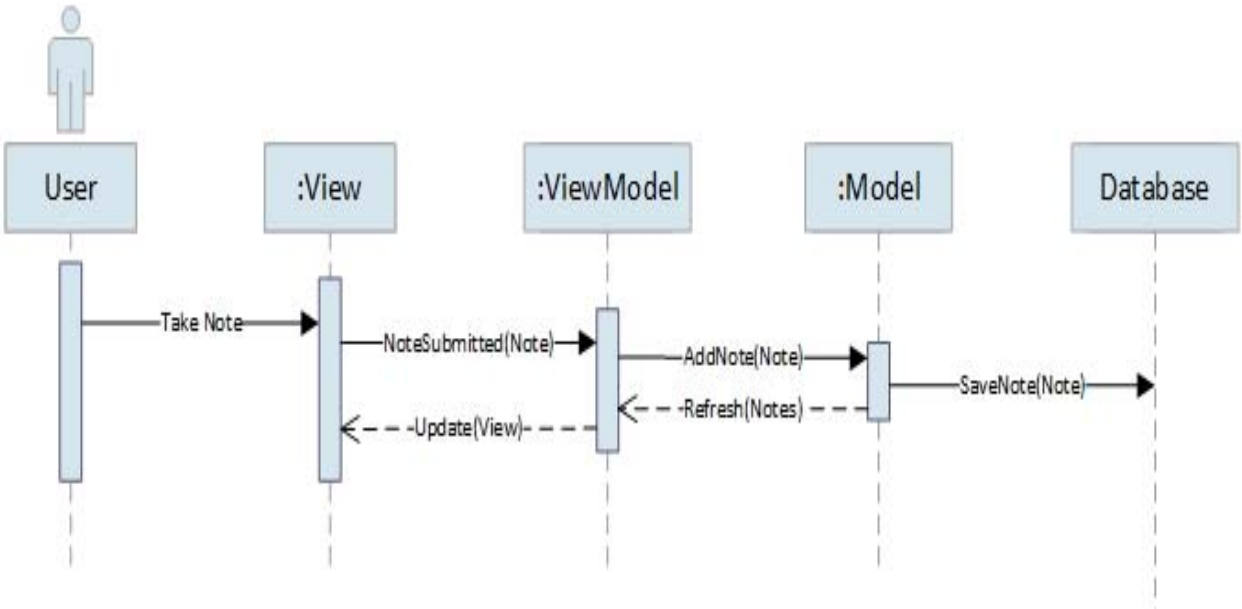

Fig 4.7 Take Note Sequence Diagram

4.3.5 Take Quiz Sequence Diagram

One of the learning-based features of In My Element is the ability for users to test their knowledge of what they have learned about the elements by quizzing themselves. Users can select elements by group (i.e. Halogens or Noble Gasses) in any combination or even all of them and then take a timed multiple choice quiz. Once the user starts the quiz, the View will notify the ViewModel where the ViewModel will then pull the information from the Model and the Database to be displayed. After the user has answered all the questions, the time will be stopped and the scores will be calculated, and the model will update the score in the database.

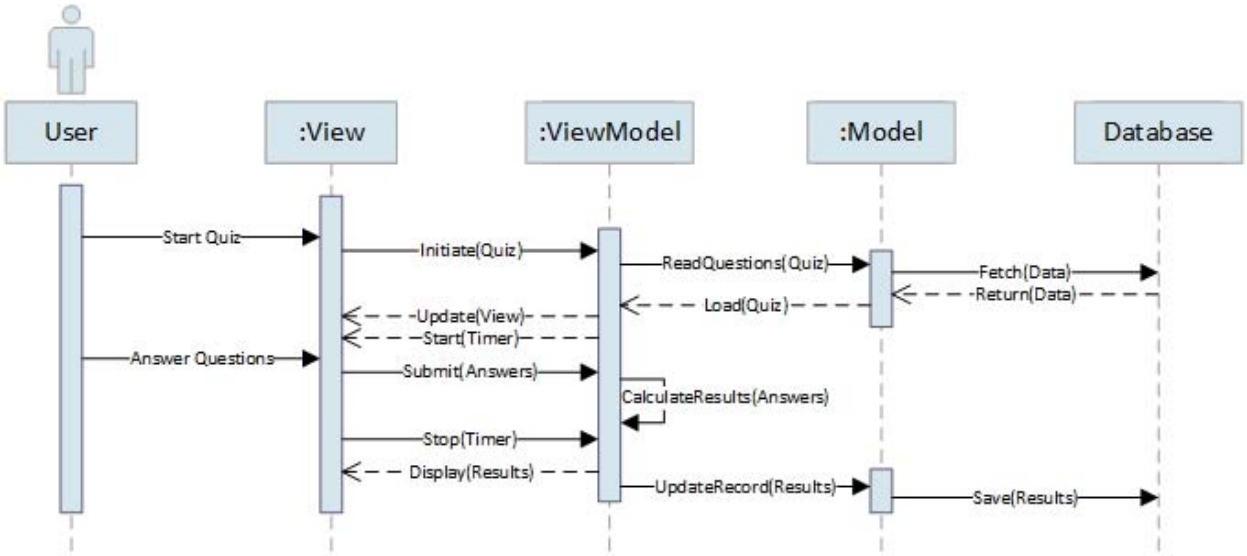

Fig 4.8 Take Quiz Sequence Diagram

4.3.6 Quiz Generation Sequence Diagram

To offer further customizability, users can generate custom quiz their own custom quiz by selecting various topics they can be tested on. Once user enters the Quiz Generation page, they are provided various check boxes that allows them to select and customize their quiz. Below is the sequence diagram displaying the interaction the View, ViewModel and Model.

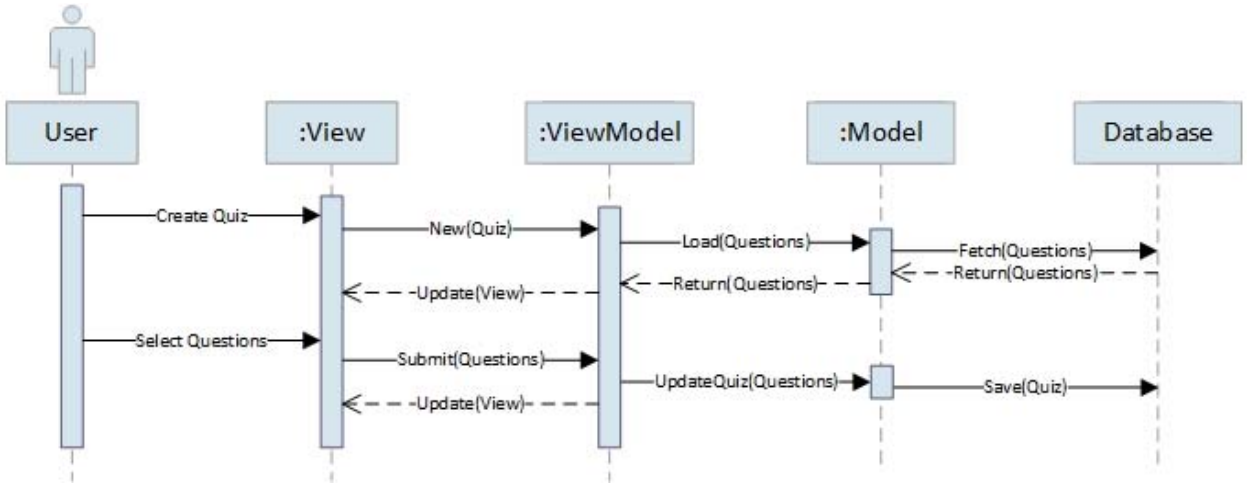

Fig 4.9 Generate Quiz Sequence Diagram

4.3.7 Puzzle Mode Sequence Diagram

The Puzzle Mode is another feature in the In My Element arsenal tool of learning tools. The Puzzle Mode allows the user to interactively fill out a blank periodic table with tiles to aid in their studies. Following the Model View, ViewModel paradigm that Microsoft Windows 8 applications use, the Puzzle Mode of this application is no different. Below shows the various classes involved when a user starts a quiz, works on, and finishes a quiz.

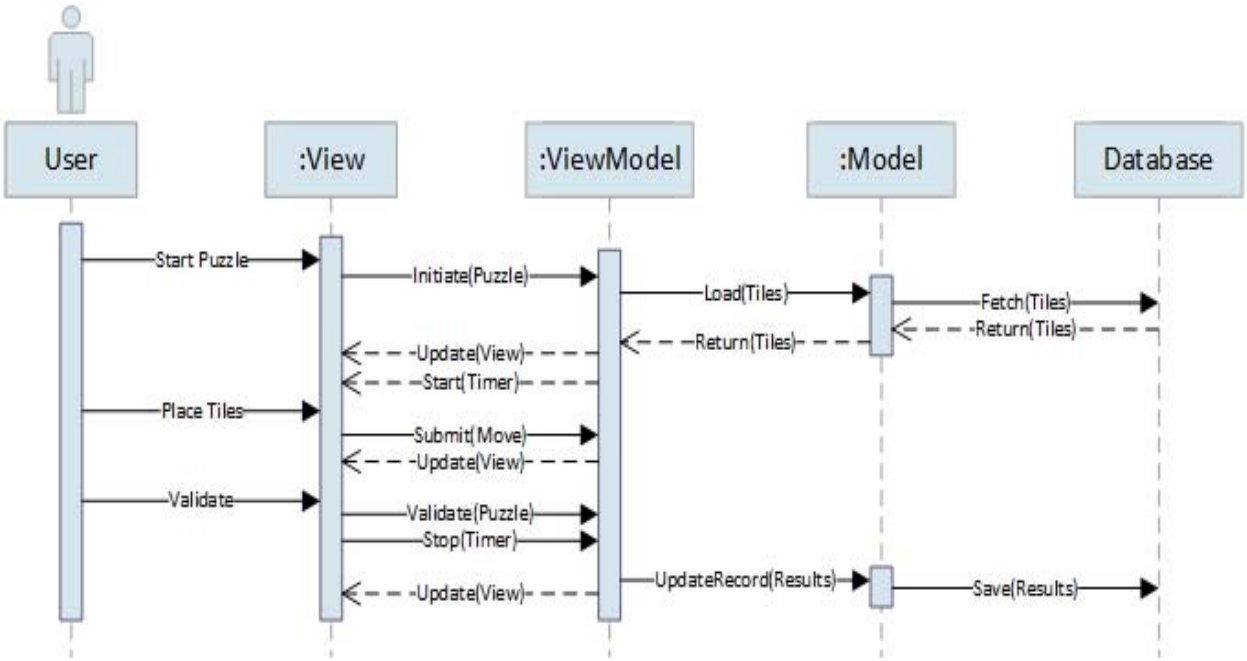

Fig 4.10 Puzzle Mode Sequence Diagram

4.4 Activity Diagrams

4.4.1 Overview

Activity Diagrams are graphical representations of the sequence of activities. It shows the workflow from the start point to the final point, detailing the many decisions and paths that exist and the program has to take to accomplish the task. Activity diagrams and sequence diagrams are similar in that they both include the various activities of the system. Sequence diagrams focus on the interaction between the classes, while activity diagrams shows the activity flow from activity

to activity. Shown below are a few activity diagrams that illustrate the flow control of In My Element from activity to activity.

4.4.2 Activity Diagram

The Activity diagram for In My Element illustrates the choices provided to the user. As soon as the application is opened, users have the option to select various modes (Study mode, Puzzle mode, or Quiz mode), which creates a fork in the internal flow of control. Once inside a specific mode, depending on the mode, users have additional options to choose from. For instance, if a user selects Study mode, they can either click on a specific element or scroll through them. They can also add notes and favorite elements from inside the Study mode UI. If a user selects Quiz mode however, they can either take a saved quiz or generate a new quiz. Each of these decisions shows a further forking of the program flow, except for Puzzle mode. There, users can begin a puzzle or return to the Home screen to choose a different option. As each of the other two options present a fork in the flow, once those paths have been traveled, the flow merges again and then again to culminate with a final choice to either exit the application or return to the Home screen. See below for the system diagram.

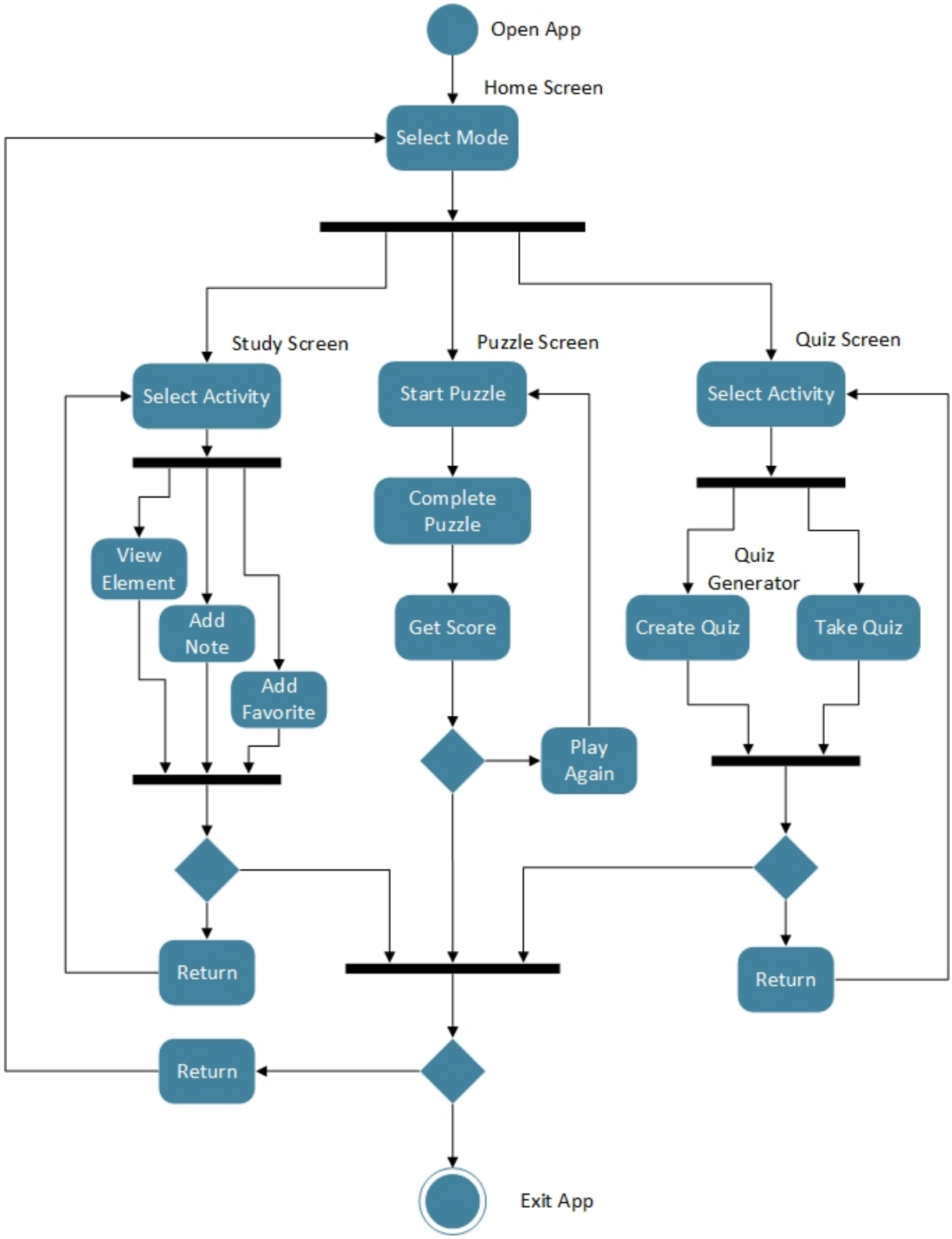

Fig 4.11 Activity Diagram

5. DESIGN AND IMPLEMENTATION

5.1 Interface Design

The In My Element Application follows the Flat design model that most Windows 8 applications have. The flat design is the stylish approach of designing the user interface without the use of 3D elements. 3D elements include things such as shadows, textures and gradients. It uses very simple graphics, typography and solid flat colors to represent the various icons, graphics and data that it needs to represent. In My Element being a Windows 8 app, the interface design was designed and developed with the flat design paradigm in mind. It features a solid black background with bright rich text and colors to represent the element. This design keeps the graphical user interface very simple, allowing younger children and older adults to easily navigate through the app. Due to the structure and time schedule of the National University capstone project, the following sections contain the screen shots of the actual user interface. The paper is due before the actual program itself, so as a result some of the screen captures look incomplete.

5.1.1 Study Mode UI Screenshot

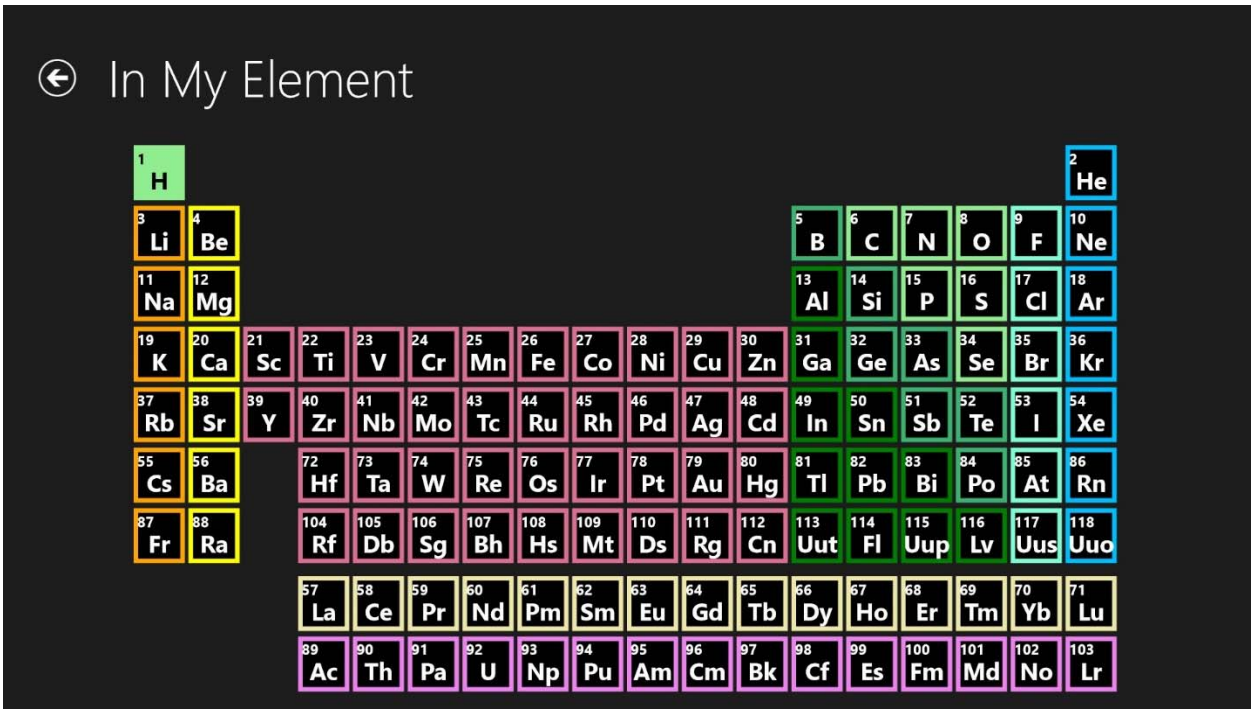

Fig 5.1 Element Selected Study Page

This a screen shot of the element Hydrogen selected on the Study Page for In My Element. Notice that the element is highlighted Green to show that it has been selected, as well as emphasizing the type of element it represents.

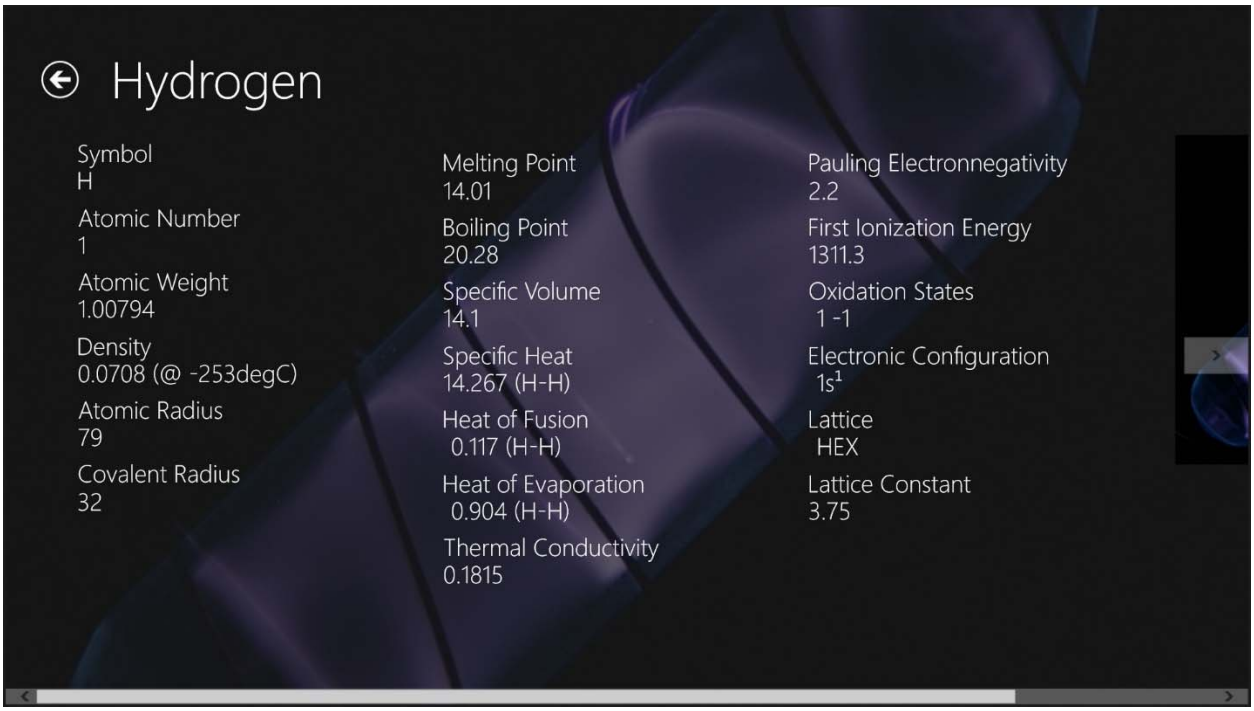

Fig 5.2 Element Selected Properties Study Page

This a screenshot of a page that pops up when an element is clicked. All the information specific to the element are detailed including a photo of the element in the background as well as in the foreground. This page is also clickable allowing you to jump to the next element that follows to see its information.

5.1.2 Puzzle Mode Screen Shot

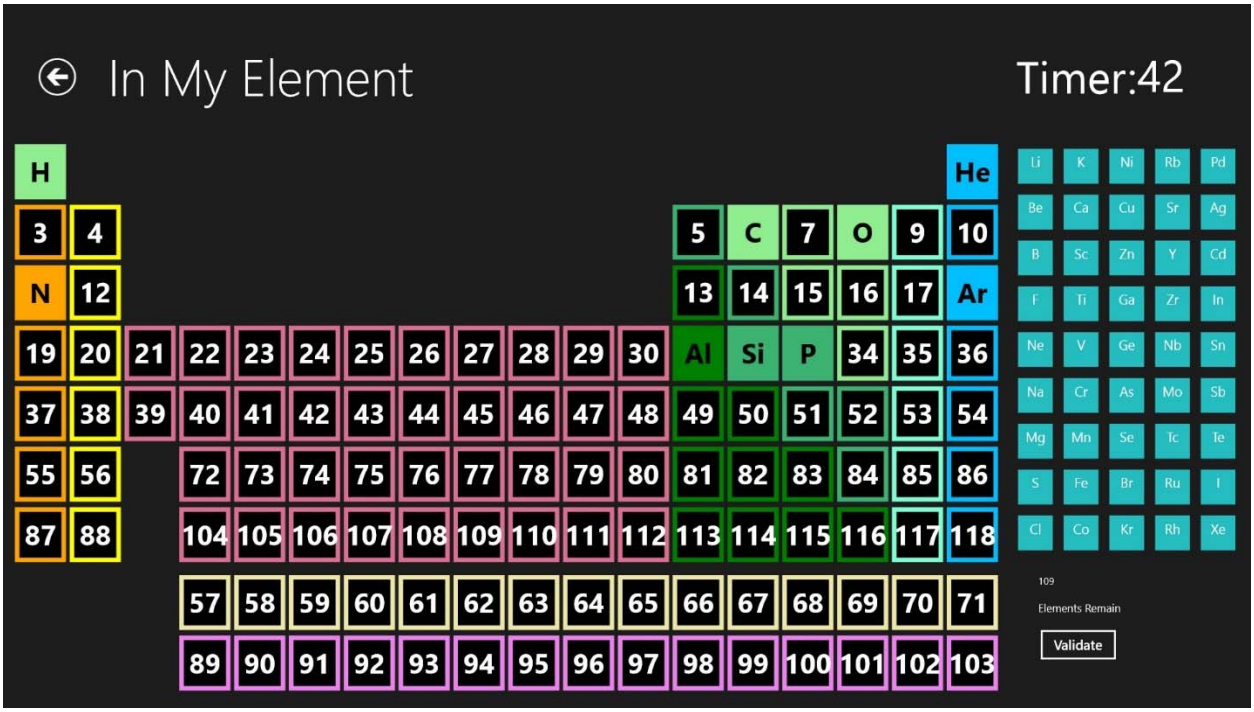

Fig 5.3 Playing Puzzle Page

Here we are actually playing a round of the Puzzle Game. It is designed to look exactly like the Study Page as far as layout, and color scheme. Here the elements are now replaced by numbers. The tiles to the left have only the element's symbol and must be matched with the elements number. A timer runs in an effort to see how quickly you can finish the puzzle. A counter below the tiles shows how many tiles are left. The validate button at the bottom right, when trigger will end the game, while giving you a score and reveal which of the tiles are in the correct position.

5.2 Network Architecture

In My Element is designed to be run on desktops, laptops, tablets, and smartphones running the Windows 8 operating system. Team KBS created an account with Windows App Store to upload and host the application on their server. It can be downloaded by any user

running the platforms listed above. As the database used is SQLite, which is an integrated database, the application does not need to reach out beyond itself to pull information in from hosted web sources. All element data is preloaded into the application.

5.3 Database Design

The following section explains the overall design of the database. It includes, the functions of the database, how it is used, the schema of the tables, and the entity resource diagram that shows the entire relationships of the database. The database of the “In My Element” application is critical to the application core. All of the element data, quiz data, and other non-user data is contained inside the database.

5.3.1 Database Diagrams

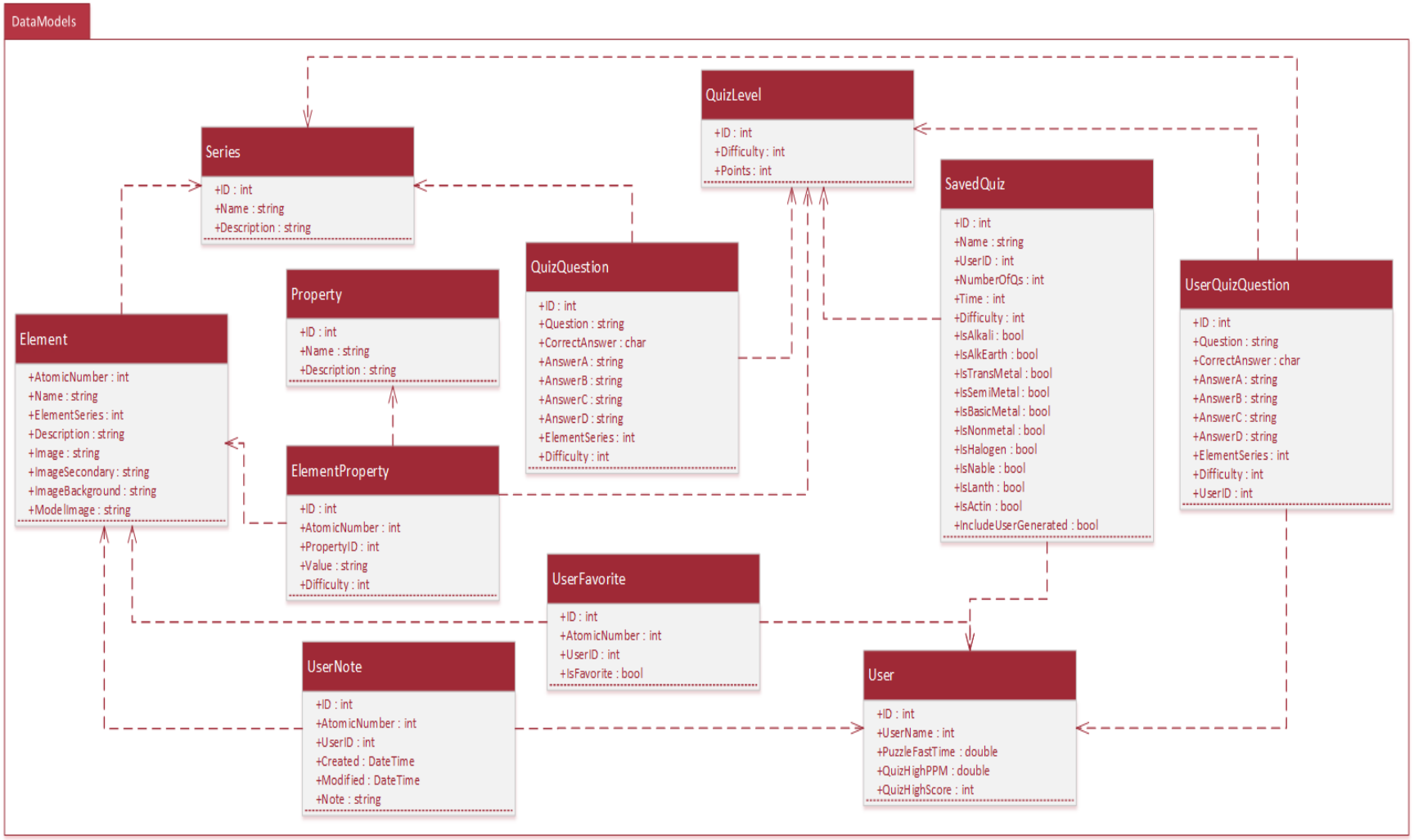

Fig 5.4 Data Models

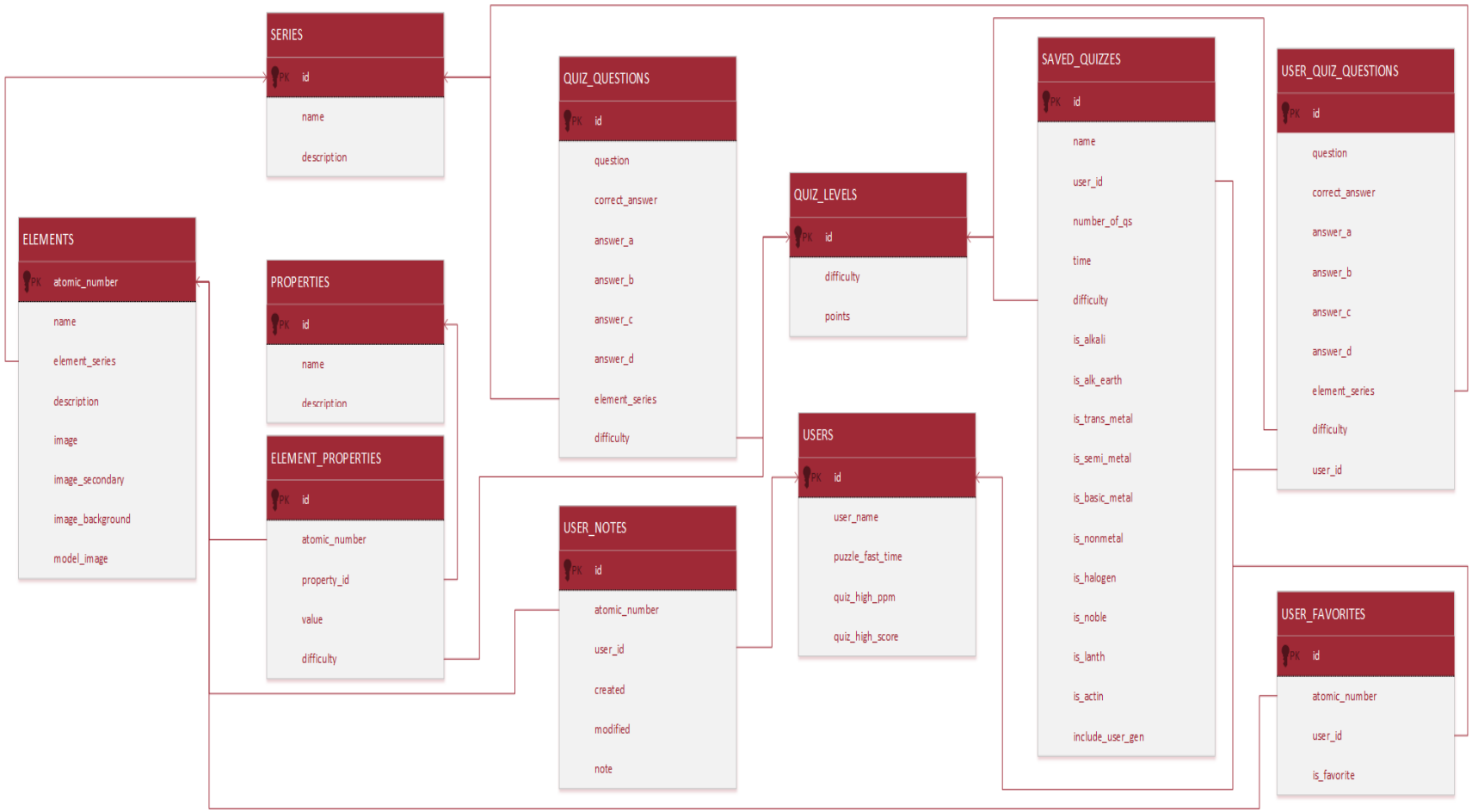

Fig 5.5 Database Design Diagram

5.3.2 Use of Database

5.3.2.1 Element Properties Population:

The Interactive Periodic Table activity of the application will provide not only basic data about an element, for instance, Atomic Mass, Element Name, and Atomic Number, but it will also provide more elaborate data, like its Crystal Structure, Specific Heat, among others. The database will house 40+ attributes for each element and multiple images. When a user clicks on an element on the table, all of this data will be on display in a scrollable textbox to the right of the table.

5.3.2.2 Quiz Question and Answer Generation:

In the Mock Quiz activity of the application, users will be able to determine the type of quiz that they want to take. They make this determination through selecting various topics that questions may be grouped. They will also determine the ease at which they progress by selecting varying levels of difficulty for the question set. The database will house questions and answers based on their topics and their corresponding difficulty (determined by the programmers). When a user first enters the Mock Quiz Activity after selecting their question set, the program will randomly display a question and a set of answers based on the User's criteria.

5.3.1.3 User Question and Answer Section:

Within the Mock Quiz, Users are granted to the ability to produce their own question and answer sets. They will be provided a section of the database that allows for placing question and answers set with a level of difficulty (determined by the user) and element series, keyed with an ID. These questions will also be accessed along with the pre-built questions in the application when Users development there question set the Mock Quiz.

5.3.1.4 Save/Load Quiz:

Once the user is done with a quiz, they will have the ability to save the quiz for their viewing pleasure at a later time. The same data that was pulled from the database, to create each question and answer set will all be saved in a section of the database designated for these saves all keyed with an ID.

5.3.1.5 User Favorites:

The Interactive Periodic Table Activity gives users the ability to favorite elements that interest them. Every time the element is shown, the icon will be highlighted if the element is set to favorite. A small portion of the database will be set to taking the ID and atomic number of the element. If there is a change of user account through the time the Application is in session, the elements that are made a favorite will be linked to that account, with a description in the database detailing whether the element is indeed a favorite.

5.3.1.6 Note Taking:

In the Interactive Periodic Table activity, Users will be given the ability to take notes on a particular element they are reviewing. A section of the database will be provided for that will take the ID and atomic number of the element. It will collect the date that it was created as well as when it was modified. Depending on which user is logged in during their session in the app, the notes themselves will be logged based on their user ID.

5.3.2 Table of Definitions

Term	Definition
PK	Primary Key - It is the unique identifier that databases have that allows the user to identify every record in the table.
FK	Foreign Key - It is the unique identifier in a relational database that identifies a row of another table. It is used to establish and enforce a link between two tables.

Table 5.1 Definitions

5.3.3 Series Table

This database table contains the element's series or the category that each element belongs to. The description of the element category is also stored here.

KEY	Attribute	Type	Description	FK Reference
PK	id	int	auto increment value	
	name	varchar(30)	The element's series (also called a category)	
	description	text	The description of the element category	

Table 5.2 Series

5.3.4 Element Table

This database table contains the properties of the individual elements. It includes information such as the atomic number, name of the element, the symbol and the description. Also stored in this table is the element associated with the individual elements.

KEY	Attribute	Type	Description	FK Reference
PK	atomic_number	int	Atomic number of element	
	name	varchar(20)	Full name of element	
	element_series	int	The element's series (also called a category)	SERIES.id
	description	text	Text description of element	
	image	varchar(30)	Location of the primary photo of the element	
	image_secondary	varchar(30)	Location of the secondary photo of the element	
	image_background	varchar(30)	Location of the image to be used as the background	
	model_image	varchar(30)	Loaction of the image of the	

			electron arrangement model	
	symbol	varchar(2)	Symbol of Element on Periodic Table	

Table 5.3 Element

5.3.5 Property Table

This database table is a more specific table that contains the name and description of the elements.

KEY	Attribute	Type	Description	FK Reference
PK	id	int	auto increment value	
	name	varchar(30)	The name of a general, atomic, physical, or other miscellaneous property	
	description	text	An explaination of the property and its meaning	

Table 5.4 Property

5.3.6 Element_Property Table

The atomic number and the element's periodic symbol and mass of one mole of the atoms of the elements are contained in this table.

KEY	Attribute	Type	Description	FK Reference
PK	id	int	auto increment value	
	atomic_number	int	Atomic number of element	ELEMENTS. atomic_num
	property_id	int	The element's periodic symbol	PROPERTIES.id
	value	varchar(30)	The mass of one mole of atoms of the element	

Table 5.5 Element Property

5.3.7 Quiz_Level

The Quiz_Level table contains the various difficulty of quizzes that the user can take. The harder the question, the more points the question is worth, thus eventually allowing a higher score.

KEY	Attribute	Type	Description	FK Reference
PK	id	int	auto increment value	
	difficulty	varchar(30)	The difficulty of the question	
	points	int	The number of points for a question at this difficulty level	

Table 5.6 Quiz Level

5.3.9 Quiz_Question

This table contains the computer generated quiz questions, not to be confused with the user generated quiz questions. It contains the question itself, the possible choices of answers, some identifiers that determine the category of elements in the questions/answers, the difficulty of the particular question. These identifiers are important because they are used to determine what questions appear at what difficulty of the quiz selected.

KEY	Attribute	Type	Description	FK Reference
PK	id	int	auto increment value	
	question	varchar(255)	Question stated to user	
	correct_answer	varchar(255)	The correct answer for the question	
	answer_a	varchar(255)	The option for answer A	
	answer_b	varchar(255)	The option for answer B	
	answer_c	varchar(255)	The option for answer C	
	answer_d	varchar(255)	The option for answer D	
	element_category	int	The category of the element in the question/answers	SERIES.id

	difficulty	int	The difficulty of the question	QUIZ_LEVELS.id

Table 5.7 Quiz Question

5.3.10 User Attributes Table

The In My Element app also keeps tracks of various high scores in the quiz and puzzle modes. This table contains the user name, the user's fastest time to complete the puzzle (in seconds), the highest number of points on a quiz (divided by the time to complete the quiz) and the user's highest score on the quiz as a percentage of the total number of points possible.

KEY	Attribute	Type	Description	FK Reference
PK	id	int	auto increment value	
	user_name	varchar(30)	The user's name	
	puzzle_fast_time	int	The user's fastest time to complete the puzzle, in seconds	
	quiz_high_ppm	float	The user's highest number of points on a quiz, divided by the time to complete	
	quiz_high_score	float	The user's highest score on the quiz as a percentage of the total number of points possible	

Table 5.8 User Attributes

5.3.11 User_Quiz_Question Table

In My Element also allows the user to generate and customize their own quiz questions. These user generated quiz questions are stored in this table. In this table, it contains the actual question that the user created, the correct answer, a few possible choices to fill in the other options in the multiple choice selection and various identifiers to identify the difficulty and elemental series of the question and the name of the user who generated the question.

KEY	Attribute	Type	Description	FK Reference
PK	id	int	auto increment value	

	question	varchar(255)	Question stated to user	
	correct_answer	varchar(255)	The correct answer for the question	
	answer_a	varchar(255)	The option for answer A	
	answer_b	varchar(255)	The option for answer B	
	answer_c	varchar(255)	The option for answer C	
	answer_d	varchar(255)	The option for answer D	
	element_series	int	The series of the element in the question/answers	SERIES.id
	difficulty	int	The difficulty of the question	QUIZ_LEVELS.id
	user_id	int	The id of the user that created the question	USERS.id

Table 5.9 User Quiz Question

5.3.12 User_Note Table

One of the unique features of In My Element is the ability for the user to generate and take their own notes about a certain element. These notes are linked to the element by the atomic number. It contains the ID of the user who created the notes, the dates that it was created and modified and the notes itself.

KEY	Attribute	Type	Description	FK Reference
PK	id	int	auto increment value	
	atomic_number	int	Atomic number of element	ELEMENTS. atomic_num
	user_id	int	The id of the user that created the note	USERS.id
	created	date	The date that the note was created	
	modified	date	The date that the note was most recently modified	
	note	text	User notes about the element	

Table 5.10 User Note

5.3.13 User_Favorite Table

The other unique feature of In My Element is the ability for the user to favorite and bookmark as many elements as they want. These favorites are associated by the atomic numbers and a Boolean logic is_favorite that marks the element as a favorite it that is set to true.

KEY	Attribute	Type	Description	FK Reference
PK	id	int	auto increment value	
	atomic_number	int	Atomic number of element	ELEMENTS. atomic_num
	user_id	int	The id of the user that created the note	USERS.id
	is_favorite	boolean	Set to "true" if the user has marked the element as a favorite	

Table 5.11 User Favorite

5.3.14 Saved_Quiz Table

In My Element allows the user to take a pre-generated computer quiz and it allows the user to make a completely custom quiz similar to a blank flash card, allowing the user to type the question and answers and word it the way they want it allowing full customization. If the user feels that they just want to take a quiz on certain properties or attributes, but not go through the process of literally writing and customizing their own question and answers, the user can check various boxes that tells the controller to include certain types of questions in the quiz. The user can name their quiz, select the number of questions and select what group or categories they want to be quizzed on.

KEY	Attribute	Type	Description	FK Reference
PK	id	int	auto increment value	
	name	int	User assigned name for the quiz	
	user_id	int	The id of the user that created the quiz	USERS.id

	number_of_qs	int	The number of questions in the quiz	
	time	int	The amount of time allowed for the quiz (in seconds)	
	difficulty	int	The difficulty level of the quiz	QUIZ_LEVELS.id
	is_alkali	boolean	true if Alkali metal questions are included	
	is_alk_earth	boolean	true if Alkaline Earth questions are included	
	is_trans_metal	boolean	true if Transition Metal questions are included	
	is_semi_metal	boolean	true if Semi Metal questions are included	
	is_basic_metal	boolean	true if Basic Metal questions are included	
	is_nonmetal	boolean	true if Nonmetal questions are included	
	is_halogen	boolean	true if Halogen questions are included	
	is_noble	boolean	true if Noble questions are included	
	is_lanth	boolean	true if Lanthanide questions are included	
	is_actin	boolean	true if Actinide questions are included	
	include_user_gen	boolean	true if the quiz includes user generated questions	

Table 5.12 Saved Quiz

5.4 Entity Relationship

5.4.1 Overview

The Entity Relationship diagram or ER diagram is a graphical representation of the entities and the relationship between the entities within a system. It is used to describe the data

aspects of the system which helps in the design of the database system. The three main components in the entity relationship diagram is the entity, the relationship and the cardinality of the relationship between the entities. An entity is the person or object of the system for example, in the In My Element app, Element is an entity. A relationship defines the activity or interaction between two entities. Cardinality is refers to the interaction between entities in terms of numbers. This section shows the entity relationship of the various entities that can be found in In My Element.

5.4.2 Entity Relationship Diagram

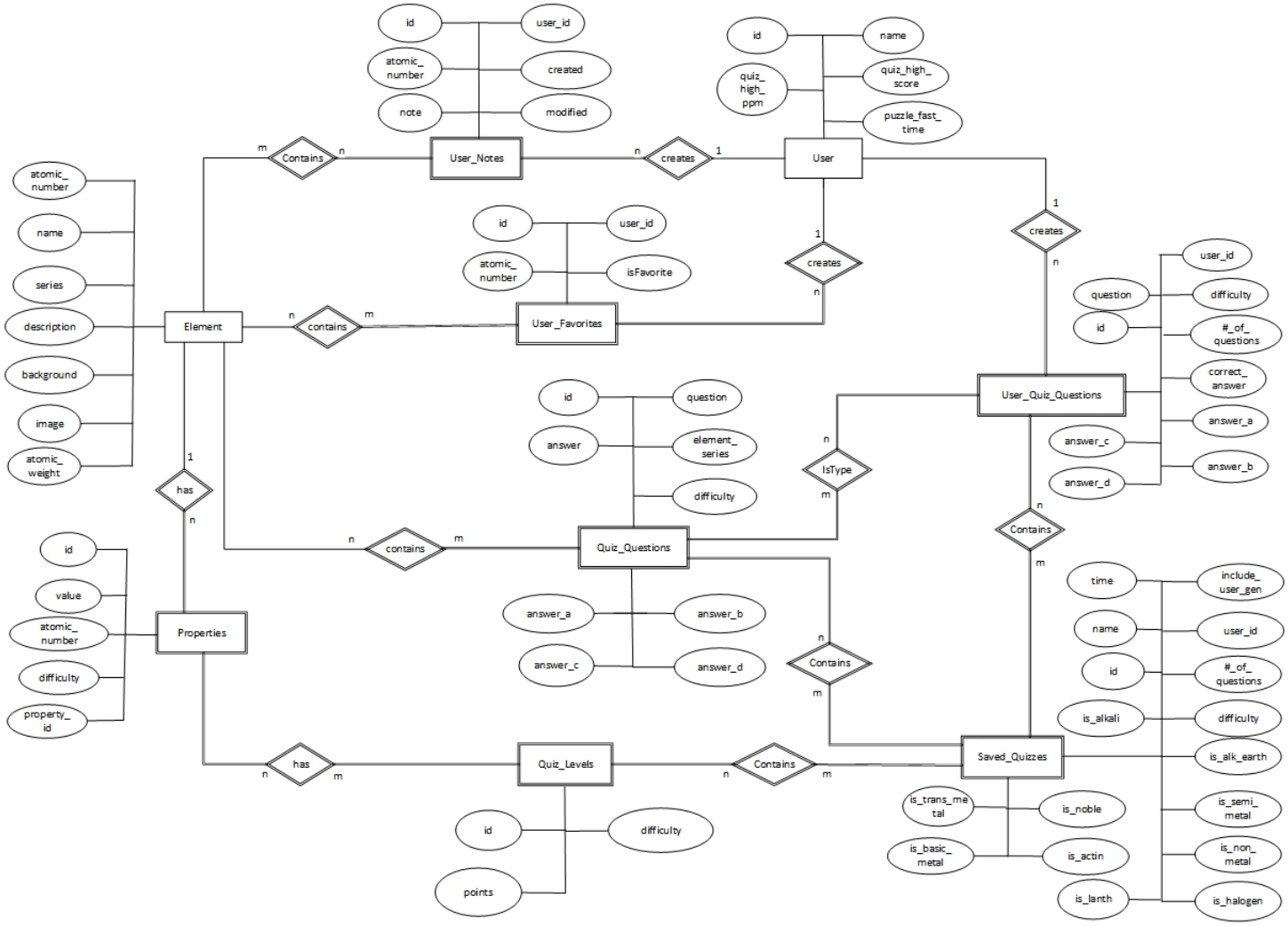

Fig 5.6 Database Entity Relationship Diagram

5.5 Database Normalization

Database Normalization is the process of organizing and break up a relational database into smaller tables to better define the relationships of the tables and to help eliminate redundancy. A database can be broken down into different normal forms. First Normal Form or 1NF is the basic form or structure of the table. It contains no repeating key variables and each row must have a primary key or the unique identifier of the particular row. Second Normal Form or 2NF will meet all the requirements of 1NF and there must not be any partial dependency of a column. Third Normal Form or 3NF meets all the requirements of 2NF and 1NF but every prime attribute in the table must depend on the primary key. There are more advance normal forms of a database such as Elementary Key Normal Form (EKNF), Boyce-Codd normal form (BCNF), Forth normal form (4NF), Fifth Normal form (5NF), Domain Key Normal Form (DKNF) and Sixth normal form (6NF) but those are beyond the scope of the In My Element Application. The database used in the In My Element application has been normalized to 3NF. One can find evidence of this in the database tables displayed in Section 5.3.2 Use of Database that it meets all the criteria of a 3NF database. It has no repeating key variables, each row has a primary key identifying the individual rows (1NF), it does not contain any partial dependency of a column (2NF) and each primary attribute in the table solely depends on the primary key (3NF).

6. TEST PLAN

6.1 Overview

In My Element is an education application for people of all ages to learn a fundamental aspect of chemistry through the interactive use of the Periodic Table of Elements. Users will be able to increase their knowledge of the elements by accessing some of the application's sub activities, either through Mock Quiz or note taking within Study Mode interactive periodic table activity. In My Element will also test a User's ability to memorize the table through recreating the table in Puzzle Mode. In the end, Users should have a firm grasp on every element and their corresponding properties to aid in their general knowledge of chemistry.

6.2 Objective and Scope

This test plan will cover the testing of the Activities that make up the In My Element Application as well as the application as a whole. This is due to In My Element being a combination of smaller individual applications, thus they will need to be tested as individual applications.

The following type tests will be conducted during each of the testing phases for each of the apps:

- Acceptance / Functional
- Integration / Systems
- GUI testing
- Load Testing
- Regression Testing

The following areas will be tested for In My Element:

- In My Element App Launch
- In My Element Application Start Screen/Activity Selection Screen UI
    - Element of the Day
    - Study Mode Activity Initiation
    - Quiz Mode Activity Initiation

    - Puzzle Mode Activity Initiation
- Study Mode Interactive Periodic Table
    - User Interface
    - Element Selection
    - Element Data Display
    - Note Taking
    - Favorite an Element
- Quiz Mode
    - User Interface
    - Quiz Difficulty Selection
    - Quiz Topic Selection
    - Quiz Question Generation
    - Quiz Answer Selection and Checking
    - Score Checking
    - Save Quiz
    - User Quiz and Database Generation
- Puzzle Mode
    - User Interface
    - Tile Generation
    - Tile Selection
    - Tile Drag and Drop Functionality
    - Puzzle Completion Check and Tile Return
- Database Access and Manipulation

6.3 Test Deliverables

The following items will be delivered upon completion of the testing.

- The complete Software Test Plan
- All test cases including Pass/Fail results
- Metrics to include test coverage and distributions

6.4 Exit Criteria

In My Element testing will consist of several exit possibilities:

1. After testing is complete, and all defects have been eliminated, testing will be considered complete and the software product can be released.
2. If software defects (at least 25%) cause testing to halt, defects will be identified and corrected, and testing will then recommence.

6.5 Personnel

The In My Element testing team consists of five National University Students; Shaun Cobb, Jordan Pan, Shane Polwort, Daari Terrell, and ChristyAnna Zimmerman. Each team member has contributed in a cooperative fashion to each test case and development to the overall project

6.6 Equipment

In order to conduct the testing of the In My Element Application, the software must be installed on a computer with Windows 8.1 as its operating system or any computer with a virtual machine installed to run Windows 8.1. Computer must have internet access for connecting to the User's Microsoft account.

6.7 Testing Schedule

This test operation will consist of multiple phases. The phase timelines are identified below.

Phases	Number of Tests	Date Due
1	8	04/28/2014
2	16	05/05/2014
3	11	05/13/2014
4	12	05/20/2014

Fig 6.1 Testing Schedule

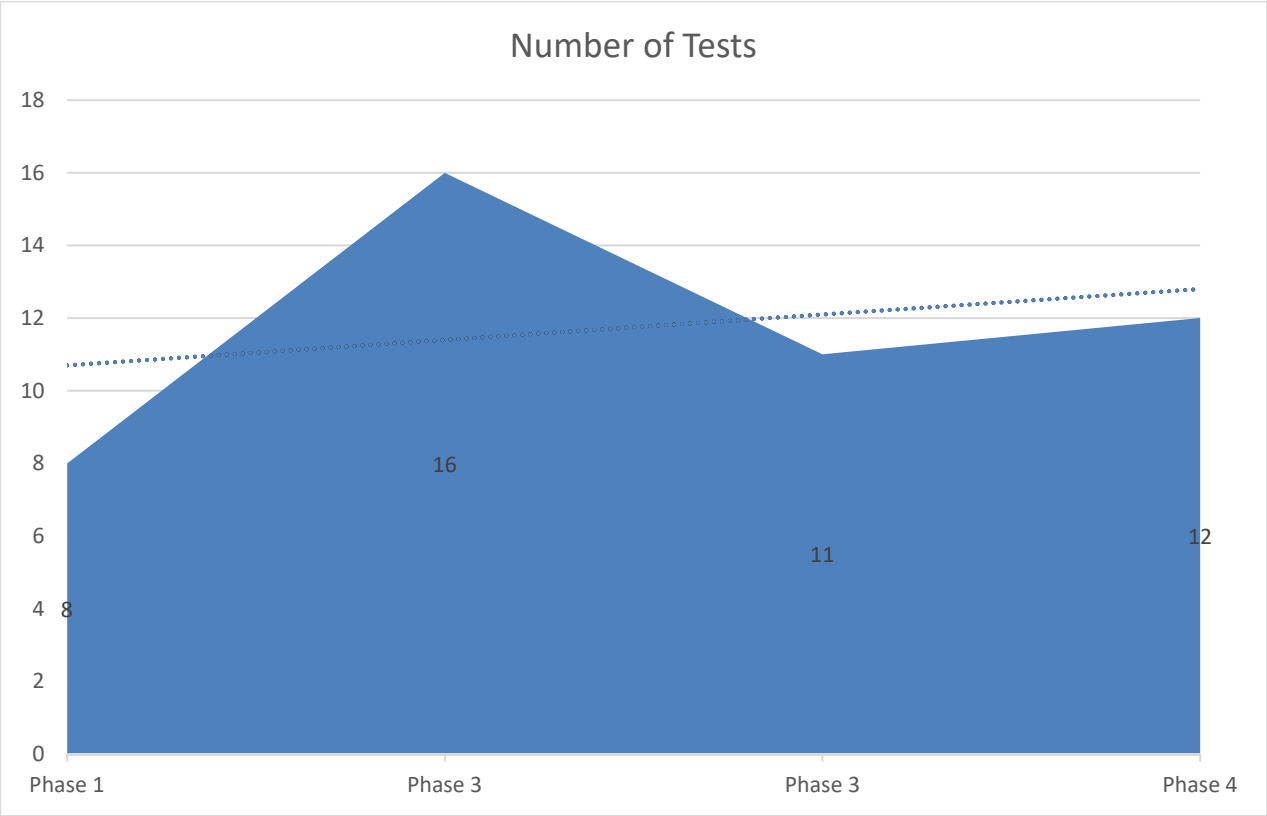

Fig 6.2 Test Cases by Phase

6.8 Test Plan Metrics

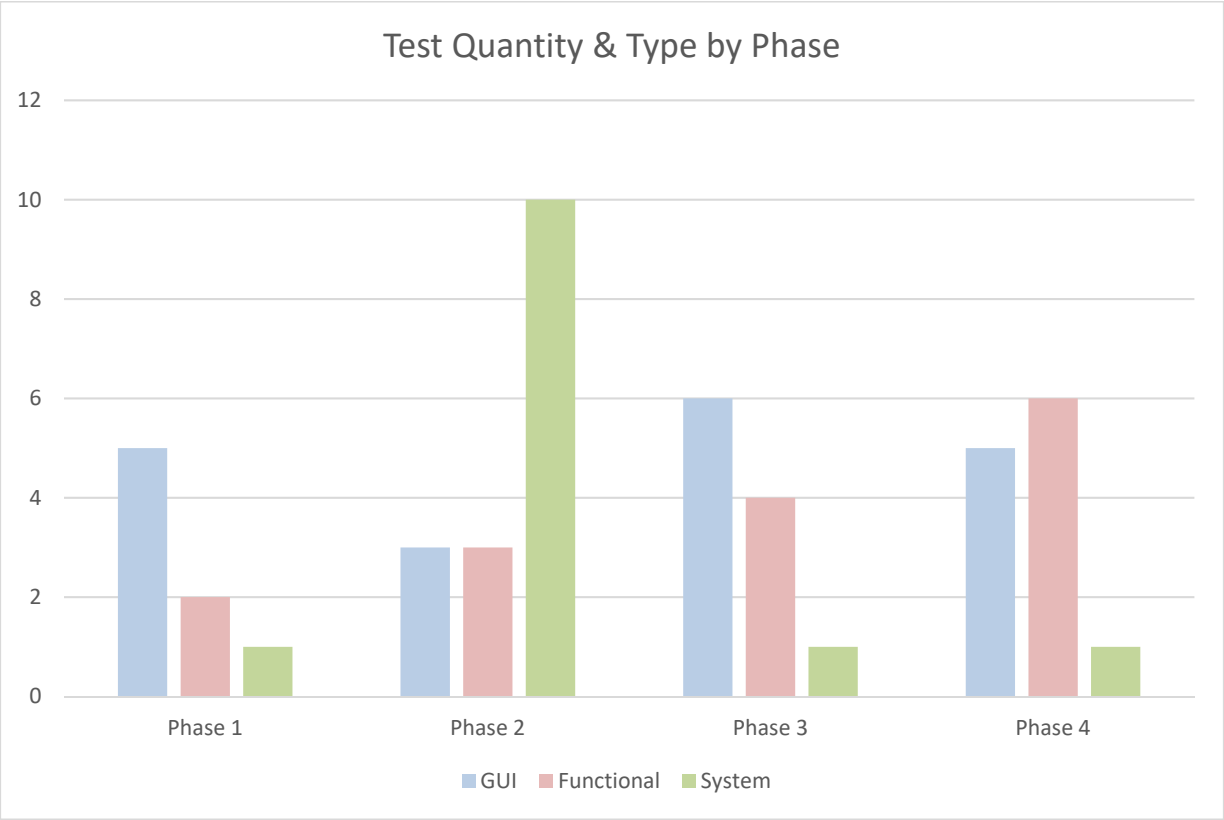

Fig 6.3 Test Plan Metrics

6.8.1 Test Types

- Acceptance / Functional Testing
- Integration and System Tests
- GUI Testing
- Regression Testing

6.8.2 Program Test Areas

This subsection is going to show the areas of the that we are going to focus on when we perform our tests as well as some of the particulars of the application that reside in these areas, as well as the type of tests that will be run. It was determined that because only a single user will open In My Element and the app doesn't connect to the internet, that load testing was unnecessary. Regression testing will typically occur when an error is found and fixed.

6.8.2.1 Start Screen App Launch

6.8.2.2 Activity Selection Screen UI

Types of testing to be performed:

- System
- UI
- Functional

Application components to be tested:

- Element of the Day Update
- Element of the Day Study Page Display
- Study Mode Activity Button
- Quiz Mode Activity Button
- Puzzle Mode Activity Button
- App Bar Functions
- Back Button

6.8.2.3 Study Mode Interactive Periodic Table

Types of testing to be performed:

- System
- UI
- Functional

Application components to be tested:

- Element Selection
- Element Data Display
- Favorite Element
- Element Notes and database generation
- App Bar Functions
- Puzzle Mode Initiation
- Quiz Mode Initiation

6.8.2.4 Quiz Mode

Types of testing to be performed:

- System
- UI
- Functional

Application components to be tested:

- Quiz Difficulty Selection Function
- Number of Quiz Questions Selection
- Quiz Topic Selection
- Quiz Generation
- Quiz Answer Generation
- Quiz Answer Selection and Checking
- Score Checking
- Skip Function
- Back Function
- Save Quiz
- User Quiz Generation and Save
- App Bar Functions

6.8.2.5 Puzzle Mode

Types of testing to be performed:

- Function
- UI
- System

Application components to be tested:

- Tile Generation
- Tile Selection
- Tile Drag and Drop
- Table Update after tile drop
- Timer
- Score Keeper
- Puzzle Completion Check

- Tile Return
- Puzzle Finish
- Table Reset
- App Bar Functions
- Back Button

6.8.2.6 SQLite Database Access and Manipulation

Types of testing to be performed:

- System

Application components to be tested:

- Element Data Access Element Data Update After Favorite
- Element Data Update after Notes
- Question Access
- Answer Access
- User Question and Answer Database Update
- User Question Access
- Saved Quiz Database Update
- Saved Quiz Access

6.8.3 Test Cases

This subsection will focus on all the test cases that we will try to cover during the testing this application. Descriptions for each test are given, along with the results we were anticipating as well as whether each test met our expectations or not. Each test will show each test type and the phase of development the test was performed.

6.8.3.1 Start Screen App Launch

Test Type		Case #	Name	Description	Expected Result	Phase
Functional Testing	1	Program Launch	Test that the application launches from the start menu	Program launches with no error	1	Pass/Fail

Table 6.1 Start Screen Test

6.8.3.2 Activity Selection Screen UI

Test Type	Case #	Name	Description	Expected Result	Phase	Pass/Fail
Functional Testing	2	Element of the Day Update	Verify that a random element is displayed on Element of the Day Section	An random element is loaded each time	1	Pass
Graphical User Interface	3	Element of the Day Study Page Display	Verify corresponding picture is loaded for element	The correct picture of the element is loaded	1	Pass
Graphical User Interface	4	Study Mode Activity Button	Verify ability to enter Study Mode	User enters Study Mode	1	Pass
Functional Testing	5	Quiz Mode Activity	Verify ability to enter Quiz Mode	User enters Quiz Mode	1	Pass
Graphical User Interface	6	Puzzle Mode Activity Button	Verify ability to enter Puzzle Mode	User enters Puzzle Mode	1	Pass

Functional Testing	7	App Bar Functions	Test App Bar Functionalities	App Bar Functionalities work properly	1	N/A
Graphical User Interface	8	Back Button	Verify back button takes user back to previous page	User goes back to previous page	1	Pass

Table 6.2 Activity Selection Test

6.8.3.3 Study Mode Interactive Periodic Table

Test Type	Case #	Name	Description	Expected Result	Phase	Pass/Fail
Functional Testing	9	Element Selection	Verify each element is selectable	Every element can be selected	2	Pass
Graphical User Interface	10	Element Data Display	Once element is selected, verify if correct information is shown	Corresponding information to element is displayed	2	Pass
Functional Testing	11	Favorite Element	Check ability to add favorites tag to each element	Able to favorite each element and is displayed on Start Menu	2	N/A
Functional Testing	12	Element Notes	Check ability to add notes to each elements	The correct notes of the elements	2	N/A
Functional Testing	13	App Bar Functions	Test App Bar Functionalities	App Bar Functionalities working properly	2	Pass
Graphical User Interface	N/A	Puzzle Mode Initiation	Verify ability to enter Puzzle Mode	User enters puzzle mode	2	Functionality Removed
Graphical User	N/A	Quiz Mode	Verify ability to enter Quiz Mode	User enters quiz	2	Functionality

Interface		Initiation		mode		Removed

Table 6.3 Study Mode Interaction Test

6.8.3.4 Quiz Mode

Test Type	Case #	Name	Description	Expected Result	Phase	Pass/Fail
Graphical User Interface	14	Quiz Difficult Selection Function	Verify each difficulty setting is selectable	All difficulty settings can be selected	4	N/A
Graphical User Interface	15	Number of Quiz Questions Selection	Verify user can select number of questions	Number of questions can be selected	4	N/A
Graphical User Interface	16	Quiz Topic Selection	Check all topics are present and selectable	All topics are present and selectable	4	N/A
Functional Testing	17	Quiz Generation	Make sure that the correct quiz is generated	The correct quiz is generated	4	N/A
Functional Testing	18	Quiz Answer Generation	Make sure that there is correct answer with generated quiz	Every quiz question has a correct answer	4	N/A
Functional Testing	19	Quiz Answer Selection and Checking	Verify correct answer matches with question	Every question has matching answer	4	N/A
Functional Testing	20	Score Checking	Verify score functionality works	Score is properly displayed	4	N/A
Graphical User Interface	22	Skip Function	Tests to see if user can skip a question	User skips to next question	4	N/A
Graphical User Interface	23	Back Function	Verify back button takes user back to previous page	Takes user back to previous page	4	N/A

Functional Testing	23	Save Quiz	Verify user generated quiz can be saved	Shows user generated quiz has been saved	4	N/A
Functional Testing	24	Load Quiz	Verify user can load a saved quiz	View shows a user generated or computer generated quiz	4	N/A
Functional Testing	25	User Quiz Generation	Verify user can generate quiz	User is able to generate quiz	4	N/A
Functional Testing	26	App Bar Functions	Test App Bar Functionality	App Bar Functionalities working properly	4	N/A

Table 6.4 Quiz Mode Test

6.8.3.5 Puzzle Mode

Test Type	Case #	Name	Description	Expected Result	Phase	Pass/Fail
Functional Testing	27	Tile Generation	Test all tiles are generated properly	All tiles are shown when screen is loaded	3	Pass
Graphical User Interface	28	Tile Selection	Test all tiles are selectable	All tiles are selectable	3	Pass
Graphical User Interface	29	Tile Drag and Drop	Test all tiles have drag and drop enabled	All times can be drag and dropped in various places on table	3	Fail
Regression Testing	30	Tile Drag and Drop	Verify that changes made to tile drag and does not affect the program as a whole.	Tile drag and drop will work Properly	3	Pass
Graphical User	31	Table Update	Verify periodic table displays tile that has been	Correct tile is	3	Fail

Interface		After Tile Drop	placed	displayed on table		
Regression Testing	32	Table Update After Tile Drop	Verify that changes made to Periodic Table do not affect the program as a whole	Correct tile is displayed in table	3	Pass
Functional Testing	33	Timer	Check functionality of timer	When timer is out of time, process ends	3	Pass
Functional Testing	34	Score Keeper	Check functionality of score	Keeps accurate score of remain tiles	3	Pass
Functional Testing	35	Puzzle Completion Check and Tile Return	Verify tile checking algorithm works	Tile checking Algorithm catches all mistakes and correct tiles	3	Pass
Graphical User Interface	36	Puzzle Finish	Verify button initiates check algorithm	Button enables tile checking algorithm	3	Pass
Graphical User Interface	37	Table Reset	Verify if button resets tiles placed on board	All tiles on table are reset	3	Fail
Regression Testing	38	Table Reset	Verify that changes made to the table reset functionality do not negatively affect the rest of the program	All tiles on the table are able reset	3	Pass
Functional Testing	39	App Bar Functions	Test App Bar Functionalities	App Bar Functionalities working properly	3	Pass
Graphical User Interface	40	Back Button	Verify back button takes user back to previous page	User goes back to previous page	3	Pass

Table 6.5 Puzzle Mode Test

6.8.3.6 SQLite Database Access and Manipulation

Test Type	Case #	Name	Description	Expected Result	Phase	Pass/Fail
System Testing	41	Element Data Access	Verify Element Data is Accessible	Element Data will be access and displayed in views of Study Mode and Puzzle Mode	2	Pass
System Testing	42	Element Data Access Element Data Update After Favorite	Verify favorite element is properly tagged	In the Element Data Table, the attribute for Favorites will be populated and then will be visible when called to the view	2	N/A
System Testing	43	Element Data Update After Notes	Verify notes are associated with proper element	In the Element Data Table, the attribute for Notes will be populated and will be visible when called to the view	2	N/A
System Testing	44	Question Data Access	Verify that data in the Questions Data Table can be accessed and viewed	Data in the Questions Data Table will be accessible, and displayed when called to the View	2	Fail
Regression Testing	45	Question Data	Verify that changes made to the accessibility of	Data in Questions	2	N/A

		Access	the Questions Data Table do not affect the rest of the program.	Data Table Will be Accessible		
System Testing	46	User Question and Answer Database Update	Verify User Created question and Answer Data is present in User Question Data Table	Data in the User Questions Data Table will be updated when queried, and displayed when called to the View	2	N/A
System Testing	47	User Question Access	Verify User Questions are accessible and viewed	User Questions will be accessible and displayed when called to the view	2	N/A
System Testing	48	Saved Question Access	Verify Saved Questions are accessible and viewed	Questions will be accessible and displayed when called to the view	2	N/A
System Testing	49	Saved Quiz Database Update	Verify newly created quiz is properly saved into database	Quiz data is stored in data table	2	N/A
System Testing	50	Saved Quiz Access	Verify a previously saved quiz is loaded	Saved Quiz data will be accessed from the Saved Quiz data table and will be displayed in the View when called.	2	N/A

Table 6.6 Database I/O Test

6.8.4 Defect Tracking

Issue ID	Date	Title	Status	Priority	Assigned Engineer	Created By
1	4/28/14	Element of the Day Update Frequency	Fixed	Low	Shaun Cobb	Daari Terrell
2	5/1/14	Missing Tiles in Tile Table after Tile Return execution	Fixed	Med	Shaun Cobb	Daari Terrell
3	5/1/14	Puzzle does not update for Elements 19 – 118 after drag	Fixed	Low	Shaun Cobb	Daari Terrell
4	5/1/14	Missing Tiles in Tile Table After Puzzle is Reset after Validation	Fixed	Med	Shaun Cobb	Daari Terrell
5	5/08/14	Database Exception	Fixed	High	Shaun Cobb ChristyAnna Zimmermann	Daari Terrell
6	5/12/14	Database Exception	Fixed	High	Shaun Cobb	Daari Terrell

					ChristyAnna Zimmermann	

Table 6.7 Defect Tracking Chart

6.8.5 Test Result Metrics

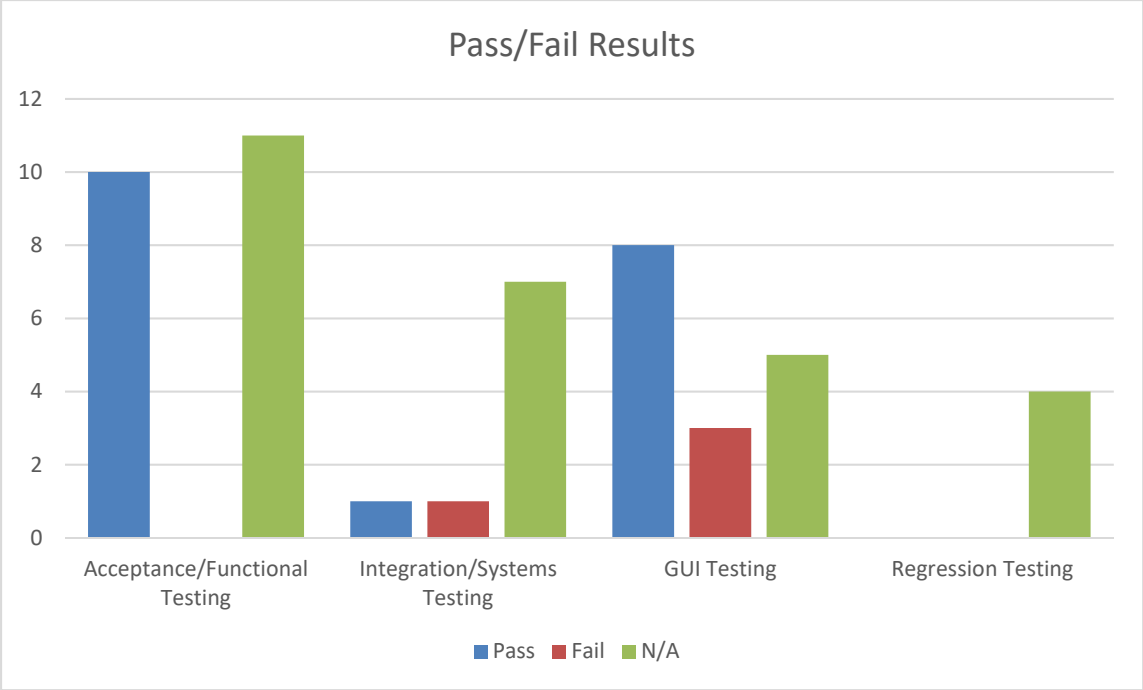

Fig 6.4 Pass/Fail Results

7. USER MANUAL WITH SCREENSHOTS

7.1 How to Download the Application and Start the Program

To download the app, the user needs to be logged into their Windows Live account and access the Windows App Store. There are two ways to access the App Store, directly from the Windows 8 "Start" menu, or by going to http://windows.microsoft.com/en-us/windows-8/apps.

1. Log into Live account and navigate to Windows 8 App Store
2. Search for "In My Element"
3. Click "Download"
4. Navigate to location of App
5. Double Click, Touch, Tap to launch app
6. Enjoy the beautiful UI
7. Scroll left and right to navigate the various modes available

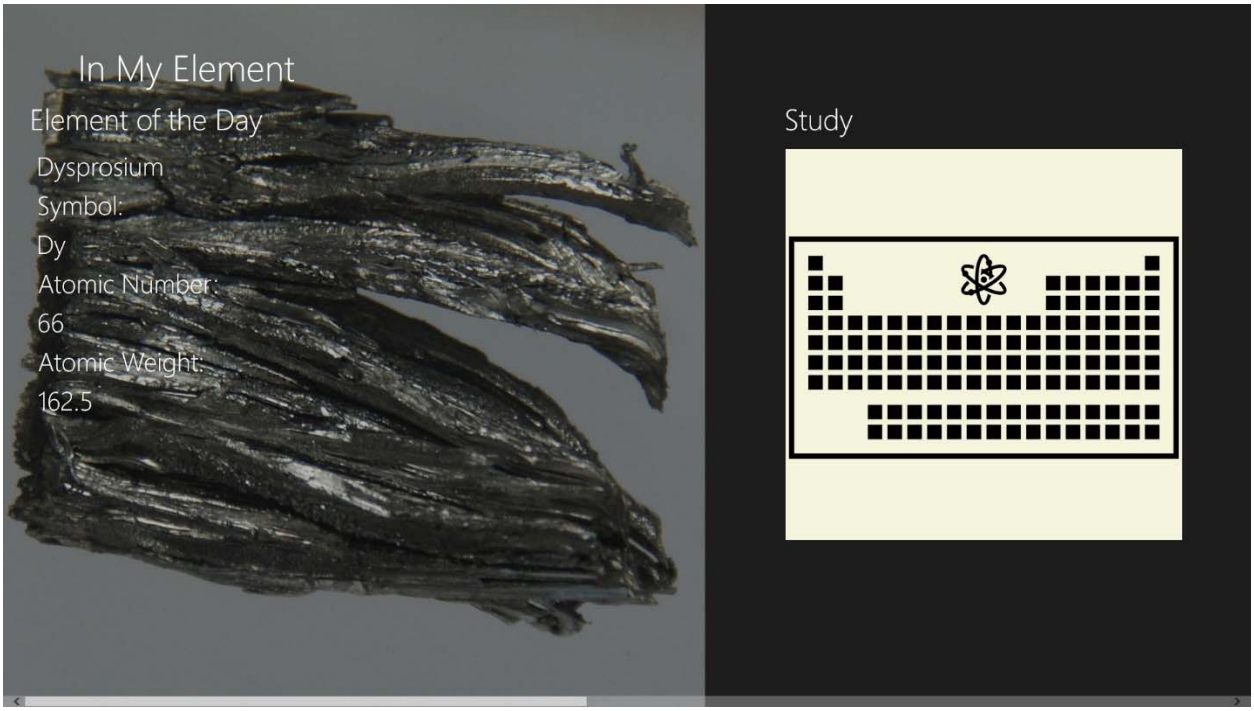

Fig 7.1 Activity Selection Screen with Element of the Day

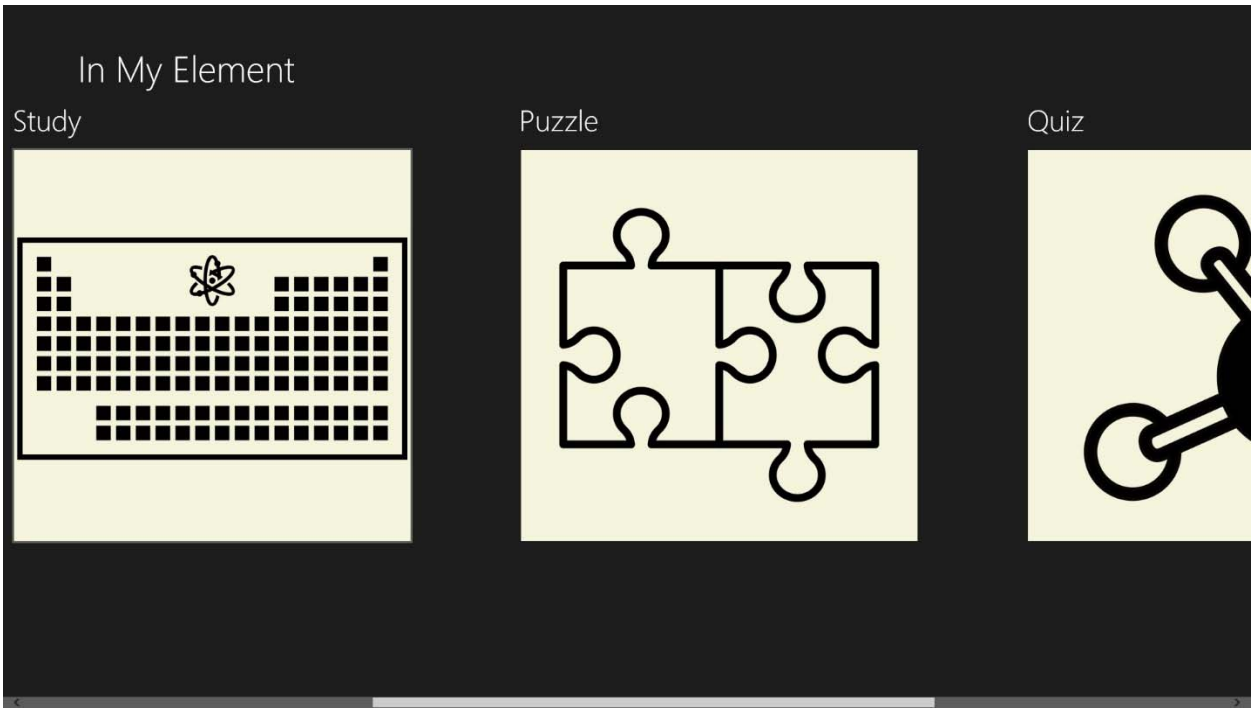

Fig 7.2 Activity Selection Screen with Study, Puzzle, Quiz Modes

7.2 How to Use Study Mode

Study Mode is the main feature of the In My Element application. It features all 118 elements on the periodic table with a very clear easy to read beautiful UI. Using the Study Mode is very intuitive. Just click or tap on the Element you want more information on, and another screen will come up displaying that information. One can close the screen and select another element, or use the scroll buttons located on the right and left side of the screen to scroll to the next or previous element based on the atomic number. From that same screen, a user can bookmark and favorite that particular element by clicking on the favorites button and they can type notes about the particular element by clicking on the notes button.

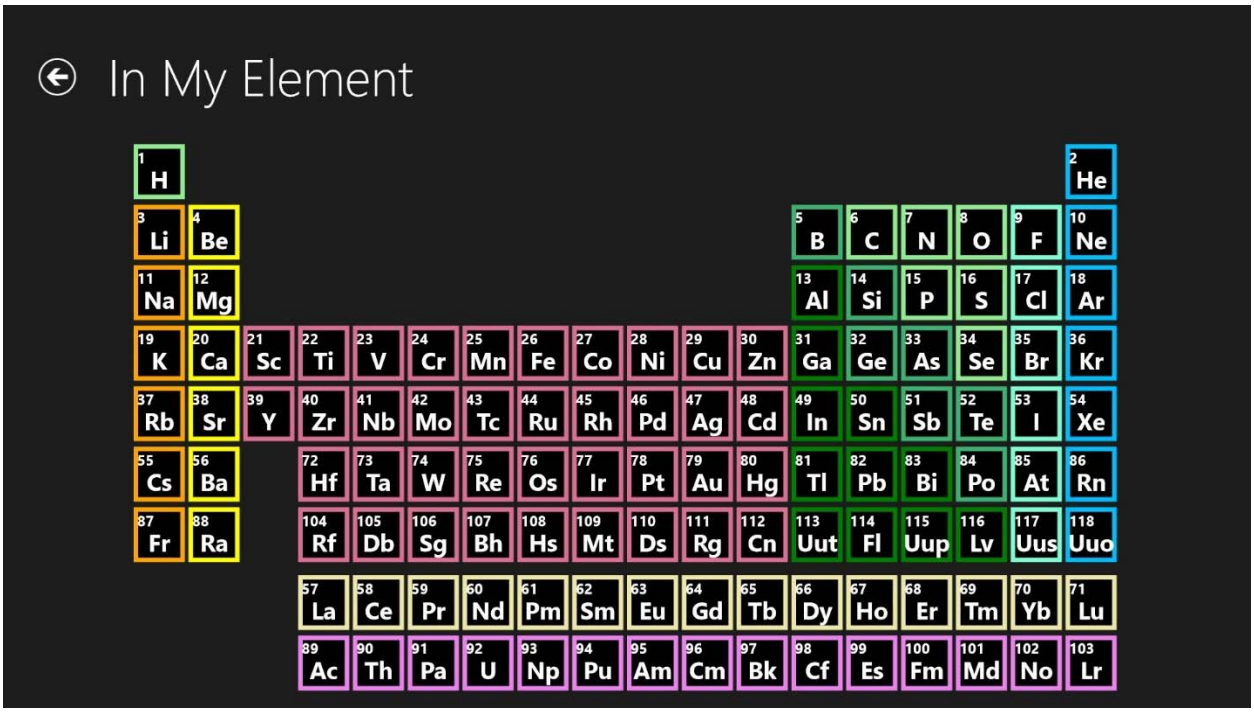

Fig 7.3 Unselected Study Mode Page

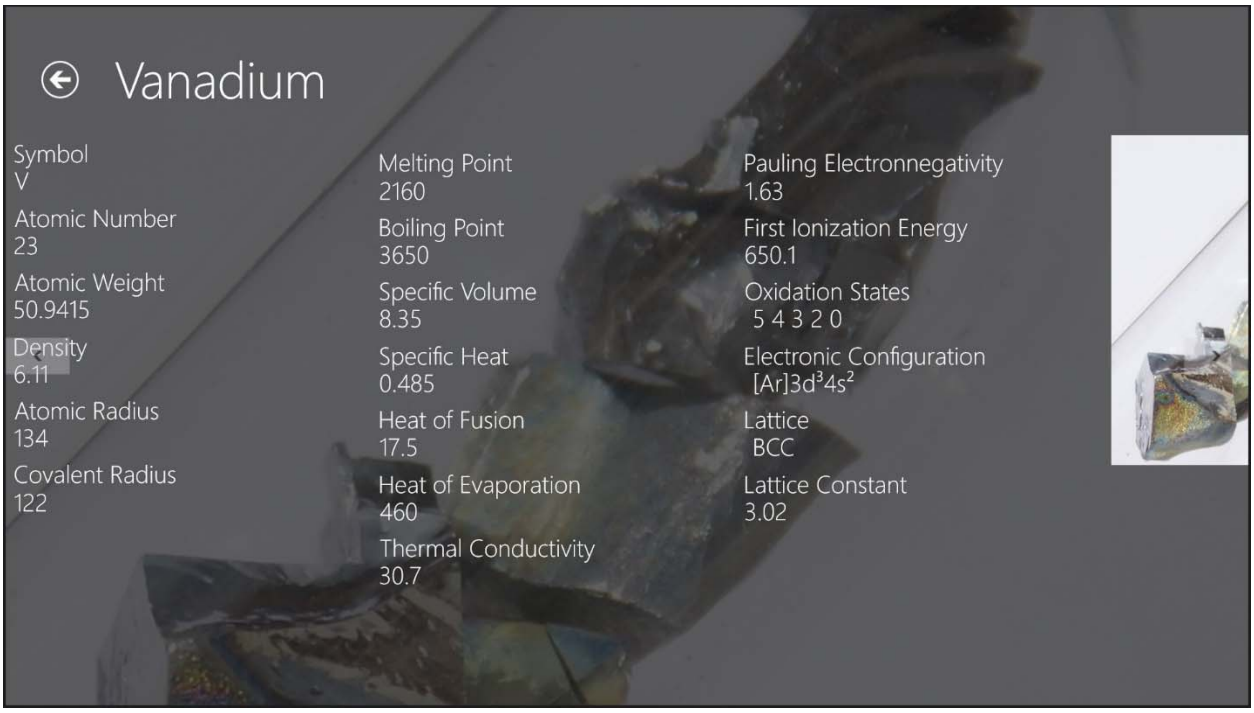

Fig 7.4 Selected Element's Properties Page

7.3 How to Use Puzzle Mode

The Puzzle mode is an interactive game that allows the user to match a list of tiles containing the atomic symbol to its atomic number. Once the user has navigated to this page, the user will need to click Start Game. Once the game starts, the timer will begin ticking and the user has to click and drag the tiles to its appropriate atomic number. Once the tile is in place, the entire square will be filled in with the appropriate color filling in the entire square. Users can remove the tiles by double clicking on the tile they want to remove and it will populate itself back into the list. After all the tiles are in place, or when the user is ready, click on Validate and the board will be checked for the right or wrong tiles. The correct tiles will have a green border around them and the incorrect tiles will have a red border around it. After all is done, the score for the particular game will be displayed.

Fig 7.5 Puzzle Mode Start

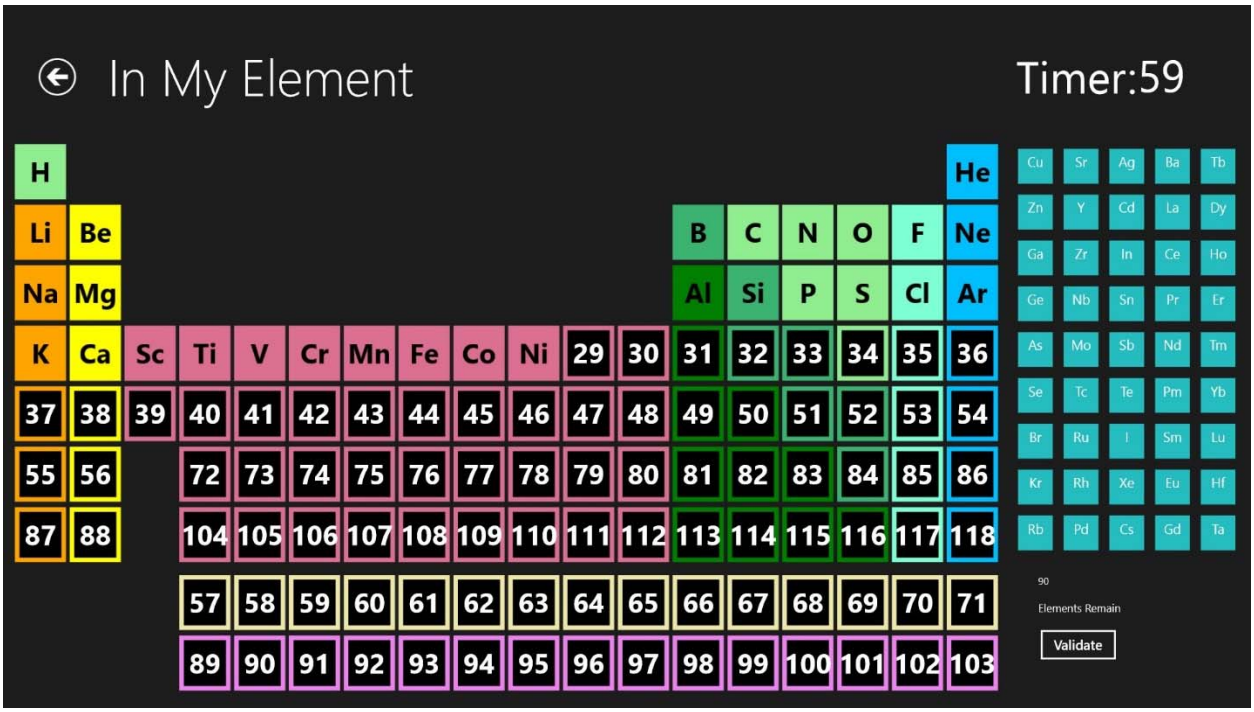

Fig 7.6 Populating the Puzzle

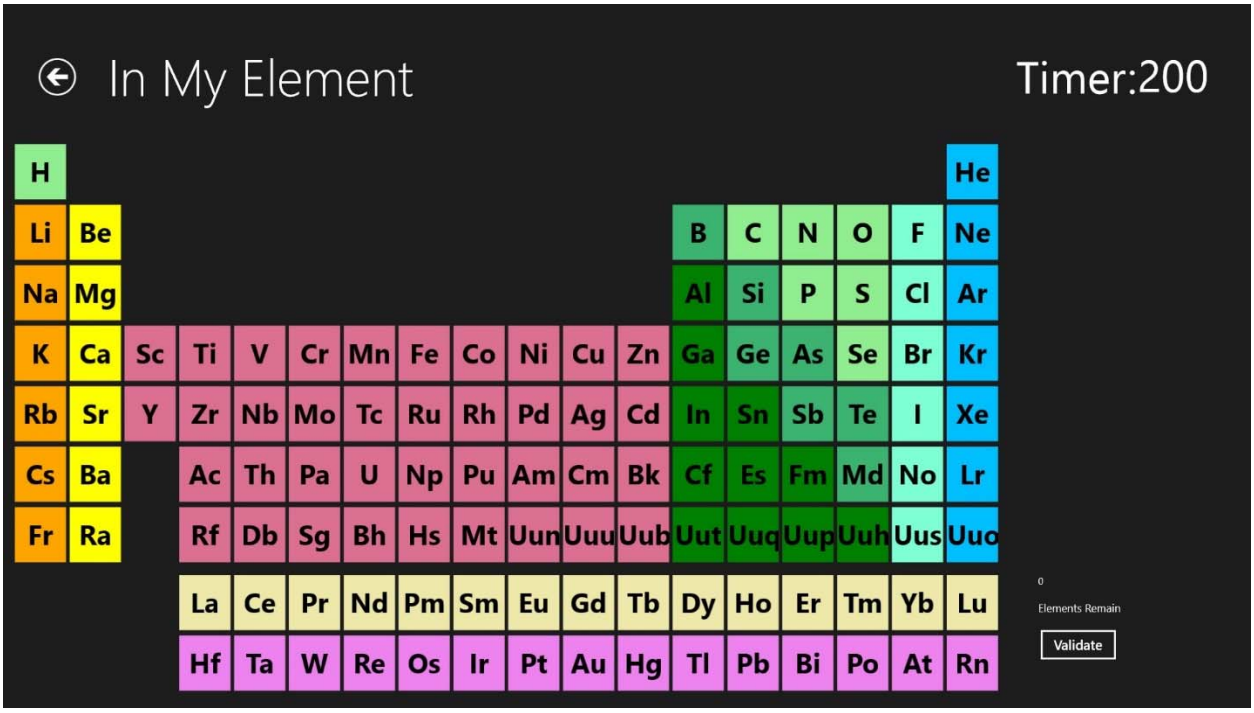

Fig 7.7 Completed Puzzle

7.4 How to Use Quiz Mode

A great way for a user to know if they understand the material is to quiz themselves. To activate quiz mode, the user will click on Quiz Mode on the main page of the app. From there they will select Easy, Medium or Hard. Once they select the difficulty settings they want to be quizzed on, the quiz will begin on a different page and the user will answer the questions be selecting A, B, C or D. After all the questions have been answered the quiz will be graded and the scores will be displayed along with the time.

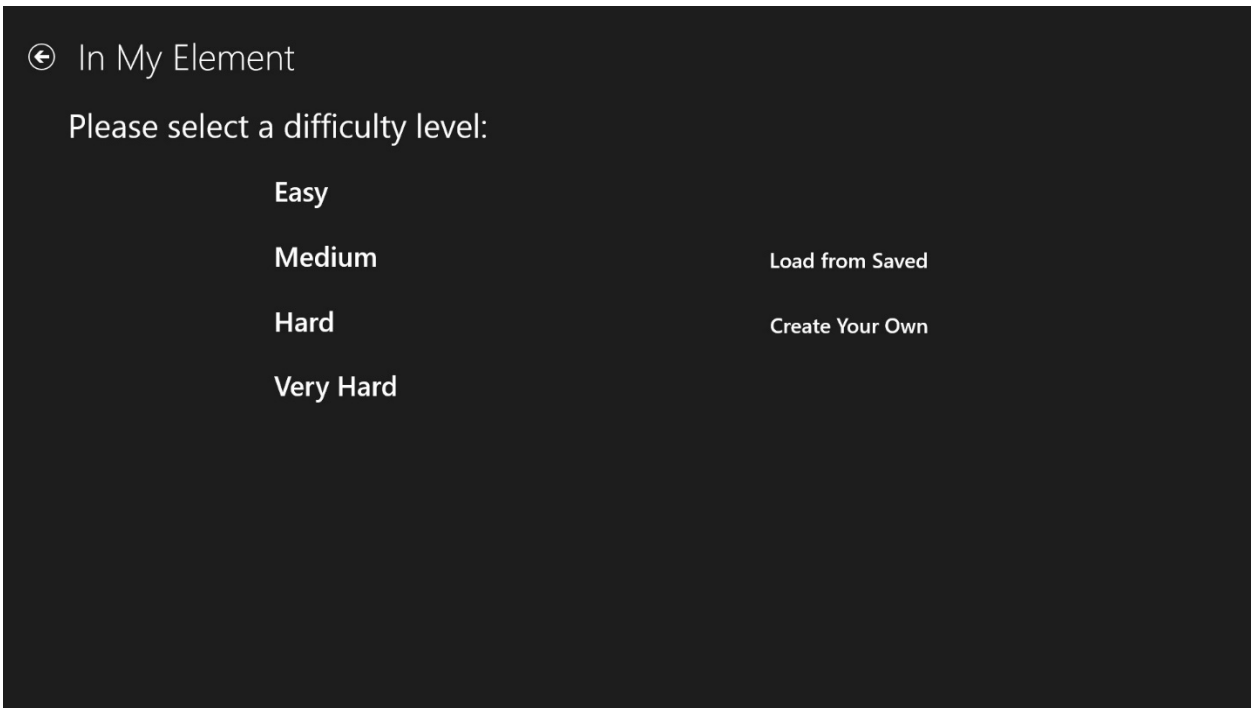

Fig 7.8 Quiz Selection Screen

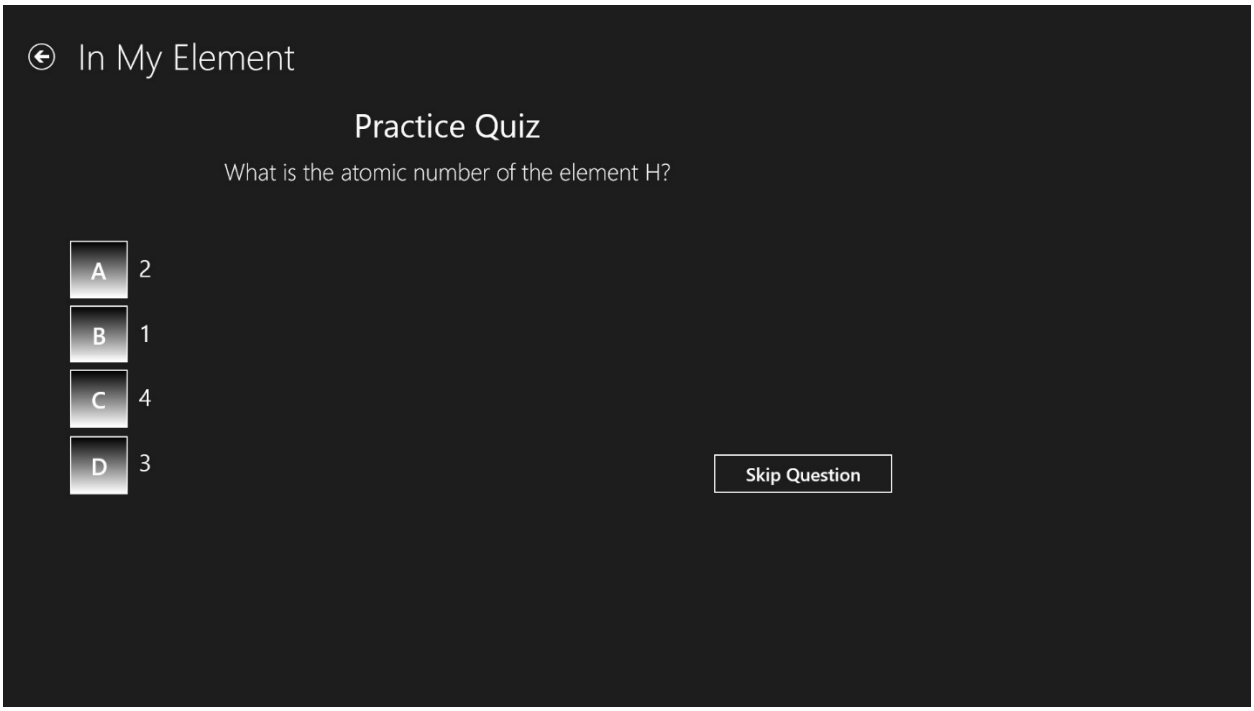

Fig 7.9 Quiz Mode

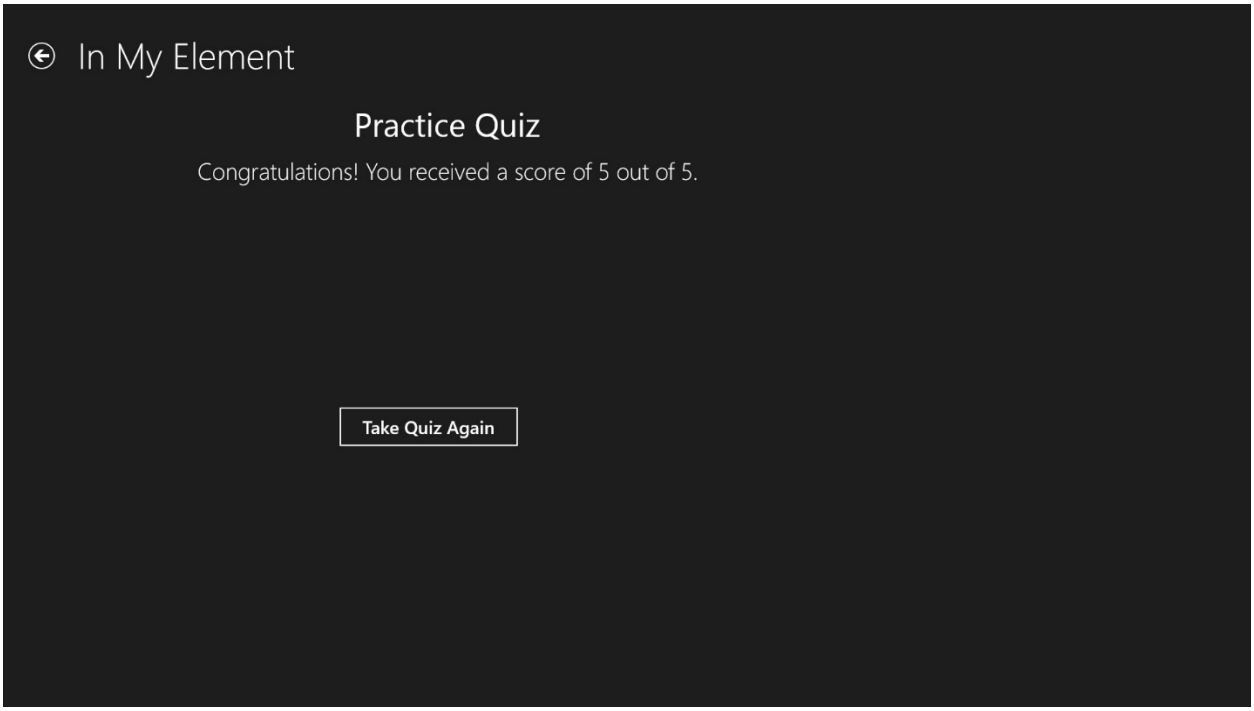

Fig 7.10 Quiz Success

7.5 How to Generate a User Quiz

To give the user further custom ability, they can generate a completely custom quiz to take. Once the user enters the Quiz Generation Mode, the user will select the number of questions, set the timer, and use the plethora of check boxes available to select the specific topics they want to be quizzed on. They can select everything from what groups, or type of elements to be quizzed on to something very specific like the number of covalent bonds only Oxygen and Carbon has. From there, the user will give a name to the quiz and add a brief comment about the quiz so they can remember what that quiz is about. Then select if they want the option to take the quiz immediately afterwards. Once all the options have been selected, the user can hit Save and the quiz will then be saved and the quiz will begin if they selected that option if not the user will be taken back to the Quiz Mode page.

8. FUTURE WORK AND CONCLUSION

As the available development schedule did not permit the team to include every feature we desired, we identified several opportunities for additional development in the future of the In My Element application.

8.1 Additional Platforms

- The application could be ported to iOS and Android so users with those platforms could take advantage of fun element-based learning

8.2 Additional Modes

- Additional quiz modes could be added to the application to increase overall learning potential for users.
- Additional game modes could be added.
    - Compound Mode
        - A mode where two or more elements are combined in specific ratios to create common chemical compounds.
    - Research Mode
        - A mode that allows users to explore the potential applications of uncommon elements or compounds

8.3 Additional Features

- The ability to print quiz results to provide a record of progression.
- Sound features to increase users' enjoyment of the application.
- Animated features to improve interactivity of the application.

8.4 Conclusion

Team KBS learned a significant amount about practical software development during the course of this project. We were compelled to research articles at sites such as MSDN and Stack Overflow to resolve problems with coding logic and syntax. We watched multitude video tutorials at sites like Pluralsight about Microsoft development technologies such as WPF and MVVM to understand how to incorporate these technologies into our application. We spent a

significant amount of time developing use cases, test cases, what-if scenarios, generating mockups and prototypes, and collaborating together to find mutually agreeable decisions about where to take the application. The environment wasn't always jovial and pleasant. There were disagreements, but ultimately we came together as a team to develop this application not only for our capstone project to complete our graduate degrees here at National University, but also for the users who will hopefully rely on our application to become better educated about the chemical periodic table and element properties in general, while doing so in an enjoyable and user friendly environment.

9. REFERENCES

Burd, S.D. (2011). *Systems Architecture* (6th ed.). Boston, MA: Course Technology, Cengage Learning.

Chemuturi, M. (2011). *Mastering Software Quality Assurance*. Fort Lauderdale, Florida: J. Ross Publishing.

Elmasri, R., & Navathe, S. (2011). *Fundamentals of Database Systems* (6th Ed.). Boston, MA: Addison-Wesley.

Lewis, W.E. (2005). *Software Testing and Continuous Quality Improvement* (2nd ed.). Boca Raton, Florida: Auerbach Publications

McConnell, S. (2004) *Code Complete: A Practical Handbook of Software Construction* (2nd ed.). Redmond, WA: Microsoft Press

Pfleeger, S., & Atlee, J. (2010). *Software Engineering Theory and Practice* (4th ed.). Upper Saddle River, NJ: Pearson

Pfleeger, C.P. & Pfleeger, S.L. (2007), *Security in Computing* (4th ed.). Upper Saddle River, NJ: Prentice Hall.

Pilone, D., & Miles, R., (2008). *Head First Software Development*. (1st ed.). Sebastopol, CA: O'Reilly Media Inc.

Pressman, R.S. (2010). *Software Engineering: A Practitioner's Approach* (7th ed.). New York, NY: McGraw Hill.

Rittinghouse, J.W. & Ransome, J.F. (2010). *Cloud Computing Implementation, Management, and Security* (1st Ed.). Boca Raton, FL: CRC Press

Rumbaugh, J., Jacobson, I., & Booch, G. (2005). *The Unified Modeling Language Reference Manual* (2nd ed.). Upper Saddle River, NJ: Addison-Wesley.

Sommerville, I (2011). *Software Engineering* (9th ed.). Boston, MA: Addison-Wesley.

10. APPENDIX A: SAMPLE CODE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using SQLite;

namespace InMyElement.DataModel {
  public class Element {
    public string Name { get; set; }
    public string Symbol { get; set; }
    [PrimaryKey]
    public int AtomicNumber { get; set; }
    public string AtomicWeight { get; set; }
    public string Density { get; set; }
    public string MeltingPoint { get; set; }
    public string BoilingPoint { get; set; }
    public string AtomicRadius { get; set; }
    public string CovalentRadius { get; set; }
    public string IonicRadius { get; set; }
    public string SpecificVolume { get; set; }
    public string SpecificHeat { get; set; }
    public string HeatFusion { get; set; }
    public string HeatEvaporation { get; set; }
    public string ThermalConductivity { get; set; }
    public string PaulingElectronnegativity { get; set; }
    public string FirstIonizationEnergy { get; set; }
    public string OxidationStates { get; set; }
    public string ElectronicConfiguration { get; set; }
    public string Lattice { get; set; }
    public string LatticeConstant { get; set; }
    private string image;
    public string Image {
      get { return image; }
      set {
```

```
            image = "/Assets/Elements/" + Name + ".jpg";
        }
    }
  }
}
```

Fig 10.1: Element.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using SQLite;

namespace InMyElement.DataModel
{
  public class QuizQuestion
  {
    [PrimaryKey, AutoIncrement]
    public int Id { get; set; }
    public string Question { get; set; }
    public string CorrectAnswer { get; set; }
    public string AnswerA { get; set; }
    public string AnswerB { get; set; }
    public string AnswerC { get; set; }
    public string AnswerD { get; set; }
    public int ElementCategory { get; set; }
    public int Difficulty { get; set; }

  }
}
```

Fig 10.2: QuizQuestion.cs

```
<UserControl
  x:Class="InMyElement.StudyElement"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
```

```
xmlns:local="using:InMyElement"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
mc:Ignorable="d"
d:DesignHeight="60"
d:DesignWidth="56">

<Grid>
  <Rectangle x:Name="ElementBox"
             Width="56"
             Height="60"
             StrokeThickness="5"
             Stroke="PaleVioletRed"
             Fill="Black"
             HorizontalAlignment="Left"
             VerticalAlignment="Top"/>
  <TextBlock x:Name="ElementNumber"
    Text="118"
    AllowDrop="True"
    FontSize="16"
    TextAlignment="Left"
    Foreground="White"
    Width="56"
    Height="60"
    VerticalAlignment="Top"
    HorizontalAlignment="Left"
    LineHeight="48"
    FontWeight="Bold"
    Padding="4,3,0,0"/>
  <TextBlock x:Name="ElementSymbol"
    Text="Uuo"
    AllowDrop="True"
    FontSize="26"
    TextAlignment="Center"
    Foreground="White"
    Width="56"
```

```
        Height="60"
        VerticalAlignment="Top"
        HorizontalAlignment="Left"
        LineHeight="48"
        FontWeight="Bold"
        Padding="0,20,0,0" />
    </Grid>
</UserControl>
```

Fig 10.3: StudyElement.xaml