Project Design Report

ADJUTOR

Shashank Khandelwal (11609001)
Shipra Aggarwal (11604838)
Shivam Chabara (11609367)
Gade Naga Manikanta Ratnasri (11602332)

Table of Contents

Abstract

The title of the application is Adjutor. The Adjutor will consist of mainly four categories like Coding calendar, News, Weather, and Wiki Search.
Coding Calendar will show the dates of all the coding events that will take place in the current month or in the coming month. The user can see all the event dates in the Coding Calendar.
The second category is the News. In this module, the user can read all the news. We can search the news by using keywords, the source who published the new.
The third category is Weather. In this, the user can know about the weather on the daily basis. The user can see the weather in his current location or in any location of his need.
The fourth category is the Wiki Search. In this user can search about anything and can get summary related to that word in a fixed number of words.

1. Introduction

It is very difficult to find everything in one application. So, we came up with our application called Adjutor. This application will make life easy.
Like it is very difficult to remember every event without any notification. So, we will be having a module called coding calendar from which the user can see all ongoing and future events and can register for the same.
The second module is weather search, this module will allow a user to get the climate condition of his current location or of a location of his choice. The weather forecast for the next five days, for the selected location will also be shown.
And the third module is news, if you want to know the updated or latest news, we have an option for news you can see the news regularly.
And the last one is Wiki search, where the user can search for any word to know the meaning of that word. And get summary related to those words in a fixed number of words.
So, the application Adjutor consists Coding calendar, News, Weather, and Wiki Search. It makes the user very easy to find everything in one application. This app will be user-friendly so that everyone can access this application easily without any problem. Adjutor contains all the information that the user needs at one place.

2. Problem Definition

In today's world everyone is busy in their own business and they are surrounded by technology. There are few things they mostly prefer to do in their spare time, while travelling or waiting, they spend time on social media and get influenced with the fake news. Fake news isn't the only issue whenever they search something on web related to their insecurities for example hair fall etc, they end up getting spams or annoying advertisements. We know that time and state of mind are always precious to us. These kind of things create bad impression on people's mind and it has been found in scientific study that all these things disturbs their state of mind and make them more aggressive towards exploring something unexpected.

What if there is a way to minimize the effect of social media and other problems discussed in above paragraph. Solution for these problems are really needed for the society and can help a lot of people.

3. Background Study/Related Work

Adjutor is a utility application and this has been already discussed in the introduction part earlier in this report, but why to make this kind of application? As we can find a lot of this kind of applications online. Reason behind making this application is to provide all those features at one place which you may easily find online but in the form of different applications which may provide some features but not all. Adjutor is a one stop shop which is completely free.

Idea of the modules inside Adjutor came from the different utility applications available online which provide few features but not all. We made a list of those applications and features they provide as well as what more we can add into Adjutor which can come handy and help people in different ways.

We came up with the modules like Weather, Coding Events Calendar, News and Wiki Search for the first release of Adjutor along with these modules Adjutor will provide Text to speech option which is right now not available in many applications.

Now, let's see how these modules will work and how we made the run. So, let's talk about the modules one by one.

3.1 Weather

Weather module took most of our time in study. While studying this module we were focused to find a valid and reliable source for weather data. We were searching for a source whose data can be moldable. Before starting this module's study we knew that Google and Yahoo have satellite for weather of their own and they both are old player in weather forecasting. So, we dig down for the search of API which they may provide and we found API from both Google and Yahoo. Point to be noted, both API don't have good support system and even Yahoo's API has been almost stopped working in case of some programming languages like python 3 for which we tested both of them. After failure of both the APIs we found another standard pypi library weather-api which was again not reliable and wasn't working as per our

expectations. Same way we tried and tested several APIs but after a lot of digging we found **OpenWeatherMap API** which is one of the most famous premium API for weather forecast but also available on open source. OWM is available for python 3 separately as **pyowm** package. At the end we tried pyowm Api and tested it as per our requirements and got reliable results.

3.2 News

News module was one of major part of owr background study as this is the module which people uses the most. Due to people use this module based applications most, our target was to find most trusted source for collecting news of almost all categories. Again we searched for key players who has vast news data from all over the world like Google, Yahoo, Bing, BBC etc. We faced two major problems, one is the data that we can get from these sources is mostly unstructured and second all the data was a mash and present in high volume. This is the fact that at this initial moment we couldn't go for our own algorithm for the segregation of this massive amount of data. So, we started search for any open source trustable API which filters out the data from the above discussed sources. As we are using python 3 for our applications backend, we knew that there must be any community supported library available for news on pypi. So, we checked the pypi and found a bunch of libraries and at that time it was difficult to test all of them one by one in order to identify which one is best or go by the ratings. We choose to go by ratings and end up getting an amazing news Api **newsapi.** We performed unit testing with this newsapi and found it useful as per our need and objectives.

3.3 Wiki Search

Wiki Search is one of the handy feature of our application. Point is why we are going to include it into our application? From our personal experience and a general survey we found that whenever people wants to know about something they usually google that thing and this thing happens in almost every case. As we all know that Google monitor our activities and keep track of everything we search on web through it's utilities. This feature of our application will prevent the misuse of people data by letting them get what they want and keep them away from google. Let's take an example to understand why staying away from google matters, suppose someone is

facing hairfall and that person searched it on google, what happens next is that person started to see annoying advertisements of hair products on web. For now this feature isn't that strong so that it can replace the need of search engine but at least it will keep people data safe at some extent. Now, let's see what we did in order to make it possible technically. This time we had only single target to search for and that was wikipedia Api and we found it very easily. After that we performed unit test with the API and found a problem with the result we got and realised that result should be more structured and counsize. So, we found the need of NLP in that case and due to NLP skills are not present in our team we forwarded NLP to the next release of Adjutor. But we found solution for NLP as well by the help of latest research going on in NLP by **stanford university.** They made an Api available for the public use but again got issue with the required skill set.

3.4 Coding Event Calendar

Coding Event Calendar module, we have especially kept for the students and those people who love to code and wants to test their skill. In our study we found that most of the students miss these coding events and end up regretting about missing them. So, we came up with an idea to add this facility in Adjutor. Now, we had to figured out how or from where we're gonna get events list. We started to search for APIs and indeed we found one named **clist.by** We tried our best to extract required data using this API but we couldn't and we tried to figure out why we couldn't able to extract data using it. In research we found that **clist.by** is a RESTful API of version 1 and it's no longer in support with python 3. Then we had to come up with something and we found a web page which contains coding events data of some of the top coding websites like Topcoders, Hackerrank, Hackerearth, Codechef, etc. Now challenge was to figure out how to extract data from that webpage and after some digging we came out with 2 ways- one using webpage automation and another is by making get and post requests for extracting HTML of that page. According to the first method we had to use selenium webdriver for web browser automation and literally we had to open the web browser get to that page and extract the html from the page but all this would happen by the help of script and user won't we able able to see any of this thing happening but due to it had a lot of dependencies we had to skip it and move towards the second way of making get and post requests for extracting HTML. Now next we had to extract the useful information out of the

extracted HTML text, basically we had to perform parsing over the text. We had to perform testing in order to get to know whether we are going to get the required information or not and also we had to understand that how parsing exactly works. We searched on web about web scraping and found a basic course on web scraping on www.Udemy.com **Web Scraping 101.** We gone through that course and understood how scraping works and successfully tested our this module.

3.5 Text to Speech

Text to Speech, we also planned to make text to speech feature available in Adjutor as we found in our study that people in today's prefer listening rather reading which encouraged us to research about this feature. In our research we found several packages available but most of them are developing state and ain't good enough to use in our application. We found most reliable package so far is by google **gTTS** (Google Text To Speech) and we performed a simple testing on it to check whether it works or not. gTTS worked and read out the written text from file.

4. Challenges

- Construction of code in a modular manner is difficult.
- Integrated testing is a bit complicated.
- Scaling and user traffic control architecture is convoluted.
- Making frontend components interact with each other becomes complex because of synchronous code.
- Difficult to decide on a particular design pattern.
- Challenging to make the app capable of reciting all news articles and everything else that the user might search for.

5. Proposed Methodology

In the proposed methodology as soon as the user launches the app, he/she will get a list for all the four option i.e. Coding calendar, news, wiki search and weather search. As soon as user will select one of the options the corresponding module will be called, and the data related to that module will be shown.

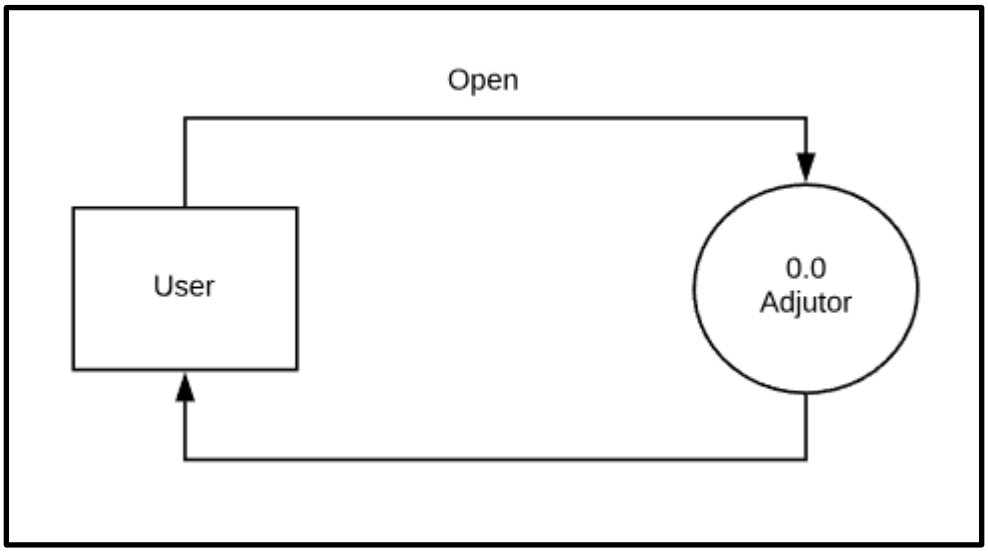
For Coding Calendar
When the user will request a coding calendar the data would be scrapped from the HTML page and will be shown to the user.

For Weather Search
When a request will be sent to this module the connection will be set with the weather API. The weather of the place will be fetched. An option for weather forecasts for 5 days will be shown and if selected the weather for the next 5 days will be fetched and will be presented.

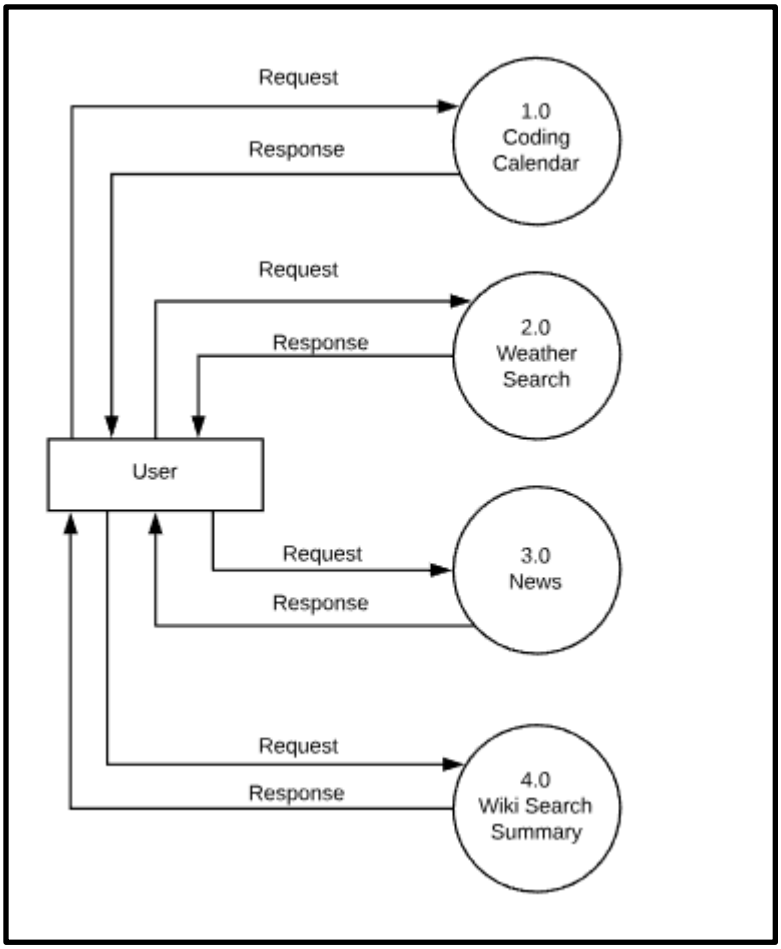
For News Search
When a request will be sent to this module connection will be established with news-API. The user will be shown the option to either search using the keyword, get the headline, get articles from a source, search the news according to date. For each of the option, the data will be fetched through API and the fetched data will be shown to the user.

For Wiki Search
When a request will be sent to this module a search bar will be shown in which user will enter the text to be searched. A connection will be made with Wikipedia API and the data for the text to be searched will be fetched. If the page will be found the user will be asked to enter the number of words in which he wants the summary, and the fetched summary will be shown. In case the page does not exist the Page not found error will be shown.

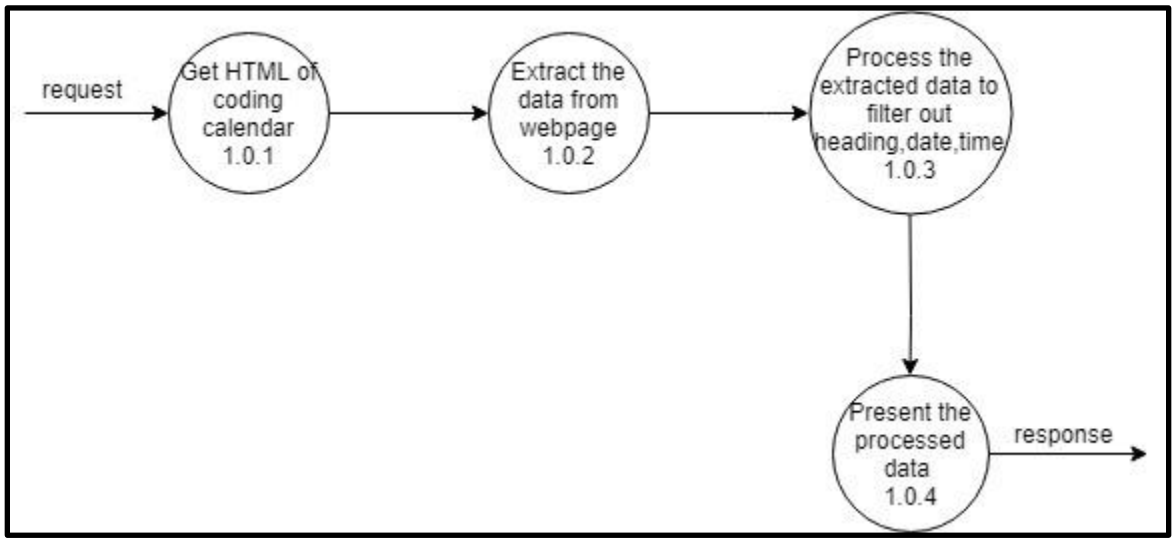
5.1 Context Level DFD



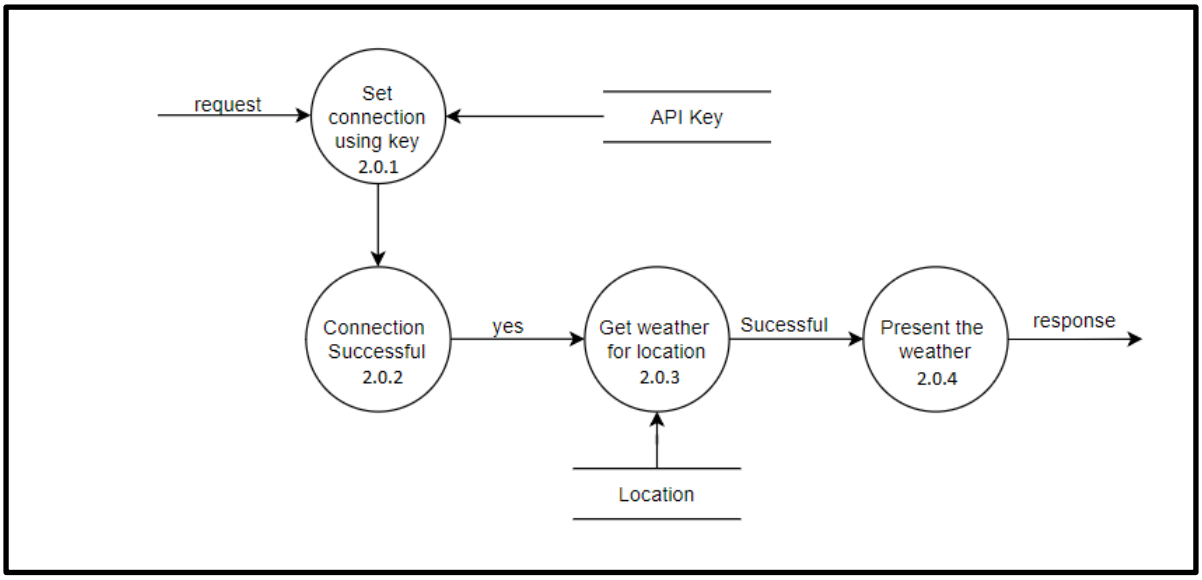
5.2 Level 1 DFD: User



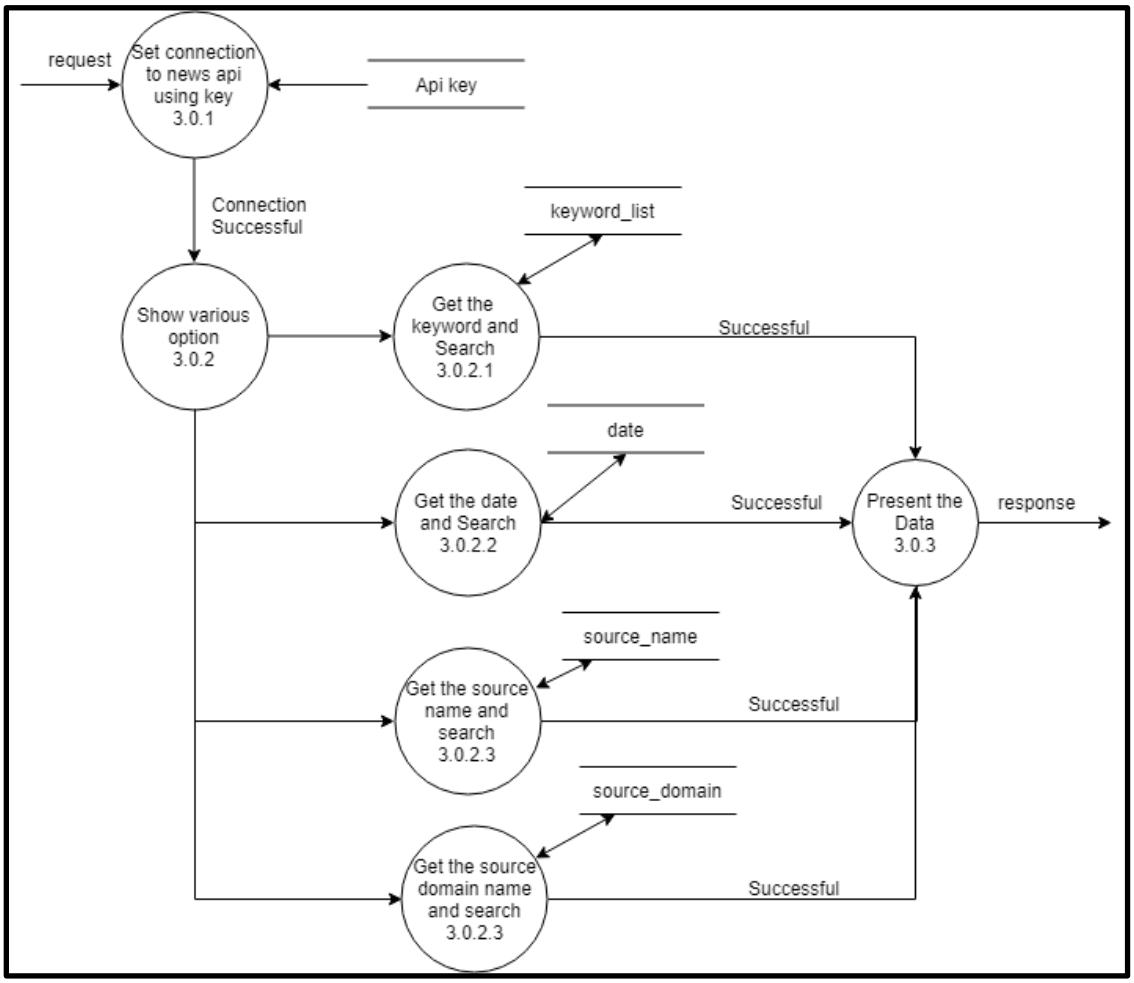
5.3 Level 2 DFD: Coding Event Calendar



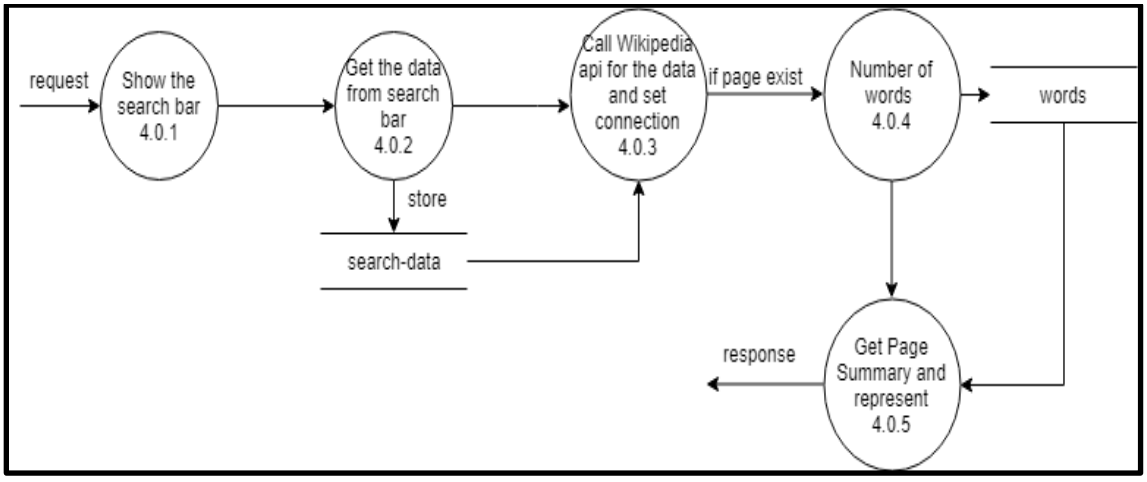
5.4 Level 2 DFD: Weather



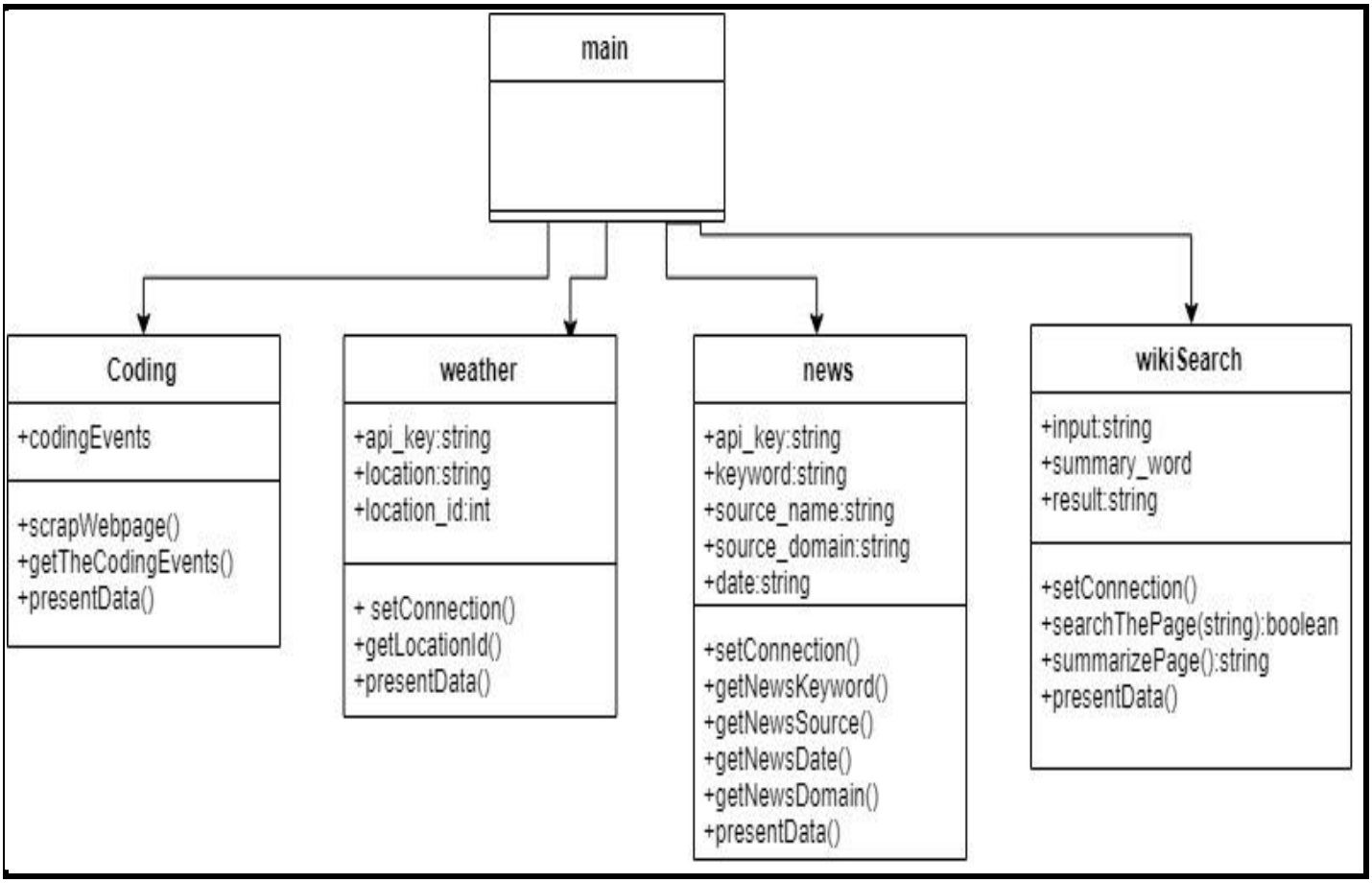
5.5 Level 2 DFD: News



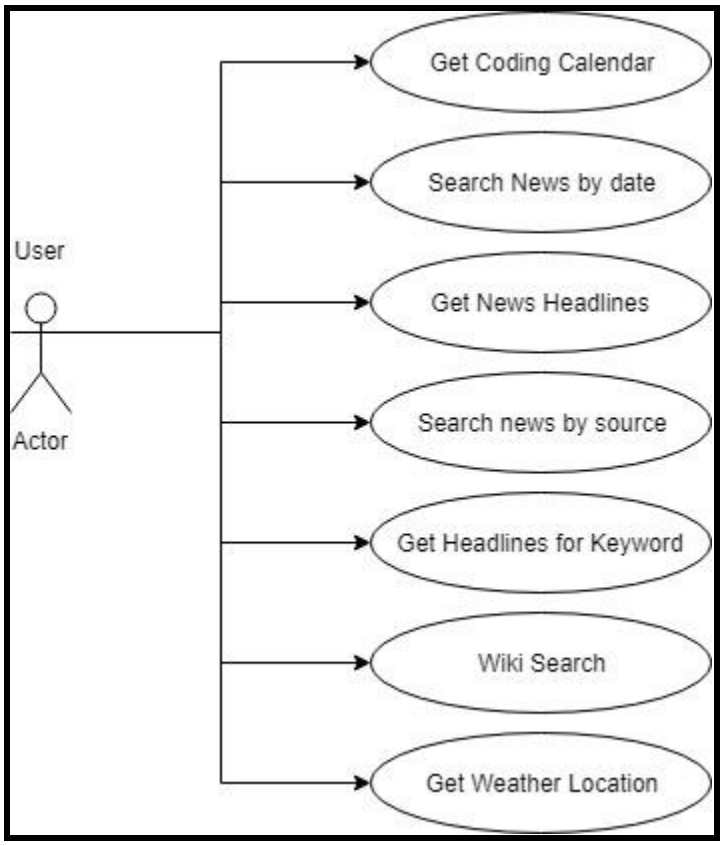
5.6 Level 2 DFD: Wiki Search



5.7 CLASS DIAGRAM



5.8 USE CASE DIAGRAM



6. Gantt Chart

	Week-1	Week-2	Week-3	Week-4	Week-5	Week-6
Understand GUI Tool						
Design GUI						
Implement Weather Module						
Implement Wiki Search Module						
Implement News Module						
Implement Coding Events Module						
Database Handling						
Implement TTS						
Integrated Testing						
Debugging						

7. Future Scope

The industry of mobile applications is worth more than 100 billion USD. In this, news apps as well as utility apps have a revenue growth rate of approximately 20%.

With millions of developers and computer enthusiasts, the competitive coding is an ever-increasing activity. People missing out on opportunities will be given a reminder to take part in competitions.

With a registered user base of more than 35 million users and 190 Billion plus views every year, a wide user base can be captured if only an app was to help the viewers find out the most important parts of the articles on Wikipedia and make it short for smoother and faster reading.

Having a variety of utilities is nothing new to the industry but no apps exist that would allow users to check out the news, the weather, lookup articles on Wikipedia and allow you to find the upcoming coding competitions at the same time.

This app can be further scaled up to include warnings of impending natural disasters and steps to be taken if users find them in such situations making it not just a utility app but one that can be relied upon in worst case scenarios.

8. References

1. https://www.hackerrank.com/calendar
2. https://clist.by/
3. https://www.codechef.com/event-calendar
4. https://openweathermap.org/api
5. https://pypi.org/project/weather-api/
6. https://pypi.org/project/yahoo-api/
7. https://code.google.com/archive/p/python-weather-api/
8. https://pypi.org/project/Wikipedia-API/
9. https://pypi.org/project/wikipedia/
10. https://newsapi.org/docs/client-libraries/python
11. https://pypi.org/project/news-api/
12. https://docs.microsoft.com/en-us/azure/cognitive-services/bing-news-search/python