POCKET HISTORY

1. **INTRODUCTION**

As the name Pocket history suggests gives you information about the remarkable historic events. Time passes by and one forgets about the events which took place. Each day unfolds a new story and each day somebody or the other across the world makes it big! Inventions, discoveries, wars, festivals, birthdays of great personalities anything and everything all at one go? Sounds great right? All you got to do is search for the date, for which you want to get the events. Get your hands on the right information fast. The application provides you with the events in the form of a list.

Most people have heard of Wikipedia. It's the ultimate repository of crowd-sourced knowledge, covering almost every subject you can think of and available to anyone with a web browser. Regardless of what information you seek, you can find it on Wikipedia, often in exhaustive detail. And because it's publicly editable, it always contains updated and relevant information.

So why this application? This is one application which will give you the list of events, be it past or present any date any year or anywhere in the world, all the big events just by typing the date and year. Every day you will be able to read what happened on this same day in history and also any date can be entered by the user according to his will. The user also has the advantage of writing a query for a particular event and get the date on which it occurred as a result.

For example:- If you enter 22 /01/1959, all the events which took place on this day across the world will be displayed.

1. **Existing System**

Search engines have made our life easier over the past few decades by giving us all the information we ever wanted. A search engine is an information retrieval system designed to help find information stored on a computer system. The search results are usually presented in links and are commonly called hits. Some of the commonly used search engines are Google, Yahoo, Bing. But when it comes to searching of dates search engines provides end number of links making it a difficult task to search for a particular event which you require.

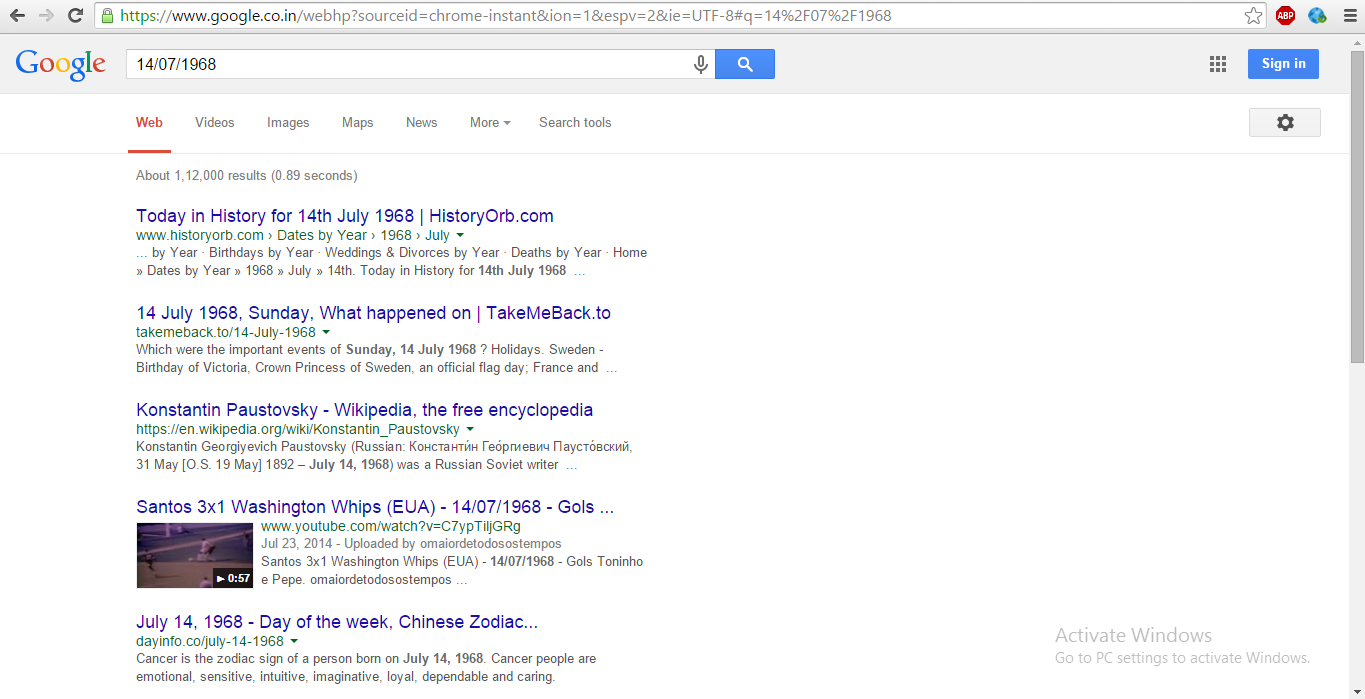
There also exists an android application known as “On this day” which displays only information for the events for a particular day for all the years .

2.1.Drawback:

1. It just displays the events for a particular day for all the years, displaying lot of data at a stretch.

2. It doesn’t give the user the control of entering a date of his own choice.

3. It doesn’t allow you to search for a date by entering the event.



1. **Proposed System**

“The only thing constant is change.” Being in the 21st century one needs to constantly upgrade themselves for better .Keeping in mind the above mentioned drawbacks the new application “Pocket History “ has been proposed.

2.1.Features:

1. It gives the user the control of entering his own customized date.

2. The user can jump to any day in history.

3. Addition to it , the user can also search using a query.

4. Provides the user with a more interactive and interesting interface.

2.2 Uses:

1.The application will help you get all the information about the events in the past and help you improve your knowledge about history.

2.General knowledge being one of the important criteria in most of the competitive exams this application will be handy for preparing for such examinations.

1. **Software Requirements Specification**

The softwares being used in our mini-project are as follows:

* 1. Android SDK:

The Android [software development kit](https://en.wikipedia.org/wiki/Software_development_kit) (SDK) includes a comprehensive set of development tools. These include a [debugger](https://en.wikipedia.org/wiki/Debugger), [libraries](https://en.wikipedia.org/wiki/Software_library), a handset [emulator](https://en.wikipedia.org/wiki/Emulator) based on [QEMU](https://en.wikipedia.org/wiki/QEMU), documentation, sample code, and tutorials. Currently supported development platforms include computers running [Linux](https://en.wikipedia.org/wiki/Linux_kernel) (any modern desktop [Linux distribution](https://en.wikipedia.org/wiki/List_of_Linux_distributions)), [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) 10.5.8 or later, and [Windows XP](https://en.wikipedia.org/wiki/Windows_XP) or later. As of March 2015, the SDK is not available on Android itself, but the software development is possible by using specialized Android applications.

3.2.Android Studio 1.2.2:

Android Studio is the official IDE for Android application development, based on IDEA.

On top of the capabilities you expect from IntelliJ, Android Studio offers:

* Flexible Gradle-based build system.
* Build variants and multiple apk file generation.
* Code templates to help you build common app features.
* Rich layout editor with support for drag and drop theme editing.
* lint tools to catch performance, usability, version compatibility, and other problems.
* ProGuard and app-signing capabilities.
* Built-in support for [Google Cloud Platform](http://developers.google.com/cloud/devtools/android_studio_templates/), making it easy to integrate Google Cloud Messaging and App Engine.

3.3.PHPStorm:

**PhpStorm** is a commercial, cross-platform [IDE](https://en.wikipedia.org/wiki/Integrated_Development_Environment) for PHP built on [JetBrains](https://en.wikipedia.org/wiki/JetBrains)'

[IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA) platform.

PhpStorm provides an editor for [PHP](https://en.wikipedia.org/wiki/PHP), [HTML](https://en.wikipedia.org/wiki/HTML) and [JavaScript](https://en.wikipedia.org/wiki/JavaScript) with on-the-fly code analysis,

error prevention and automated [refactorings](https://en.wikipedia.org/wiki/Refactoring) for PHP and JavaScript code. PhpStorm is built

on IntelliJ IDEA, which is written in [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). Users can extend the IDE by installing plugins created

for the IntelliJ Platform or write their own plugins.

3.4.Ubuntu:

**Ubuntu**  is a [Debian](https://en.wikipedia.org/wiki/Debian" \o "Debian)-based [GNU/Linux](https://en.wikipedia.org/wiki/GNU/Linux) [operating system](https://en.wikipedia.org/wiki/Operating_system) and [distribution](https://en.wikipedia.org/wiki/Linux_distribution), with [Unity](https://en.wikipedia.org/wiki/Unity_(user_interface)) as its

default [desktop environment](https://en.wikipedia.org/wiki/Desktop_environment) for [personalcomputers](https://en.wikipedia.org/wiki/Personal_computer" \o "Personal computer) including [smartphones](https://en.wikipedia.org/wiki/Smartphone) in later versions.

Ubuntu also runs [network servers](https://en.wikipedia.org/wiki/Network_servers). It is based on [free software](https://en.wikipedia.org/wiki/Free_software) and named after the Southern

African philosophy of ubuntu which often is translated as "humanity towards others" or "the

belief in a universal bond of sharing that connects all humanity".

3.5.ArgoUML:

ArgoUML is an [UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language) diagramming application written in Java and released under the open

source [Eclipse Public License](https://en.wikipedia.org/wiki/Eclipse_Public_License). By virtue of being a [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) application, it is available on any

platform supported by Java.

3.6.Ruby script:

Ruby is a language of careful balance. Its creator, [Yukihiro “Matz” Matsumoto](http://www.rubyist.net/~matz/), blended

parts of his favorite languages including Perl, Smalltalk, Eiffel, Ada, and Lisp to form a

new language that balanced functional programming with imperative programming.

In Ruby, everything is an object. Every bit of information and code can be given their own

properties and actions. Object-oriented programming calls properties by the name instance

variables and actions are known as methods. Ruby’s pure object-oriented approach is most

commonly demonstrated by a bit of code which applies an action to a number.

3.7 Genymotion:

Genymotion is an Android emulator for building and testing great Android apps. It’s fast,

simple and powerful.

It offers 20 pre-configured devices and you can create your own custom ones.

3.8 Github:

**GitHub** is a web-based [Git](https://en.wikipedia.org/wiki/Git_(software)" \o "Git (software)) repository hosting service, which offers all of the distributed

revision control and [source code management](https://en.wikipedia.org/wiki/Source_code_management) functionality of Git as well as adding

its own features. Unlike Git, which is strictly a [command-line](https://en.wikipedia.org/wiki/Command-line) tool, GitHub provides a web-

based graphical interface and desktop as well as mobile integration. It also provides access

control and several collaboration features such as [wikis](https://en.wikipedia.org/wiki/Wiki), [task management](https://en.wikipedia.org/wiki/Task_management), and bug

tracking  and [feature requests](https://en.wikipedia.org/wiki/Software_feature) for every project.

1. **Hardware Requirements**

Memory: 4GB

HDD:500GB

CPU Intel Core i3-4030U,1.9GHz

1. **Software Requirements**

1.Android SDK

2. Android Studio 1.2.2

3. PHPStorm

4.Ubuntu

5.Ruby programming language

6.Genymotion

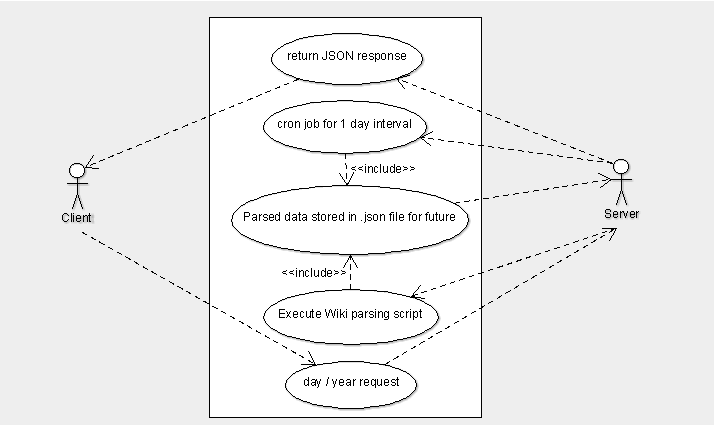
7.GitHub

8.ArgoUML

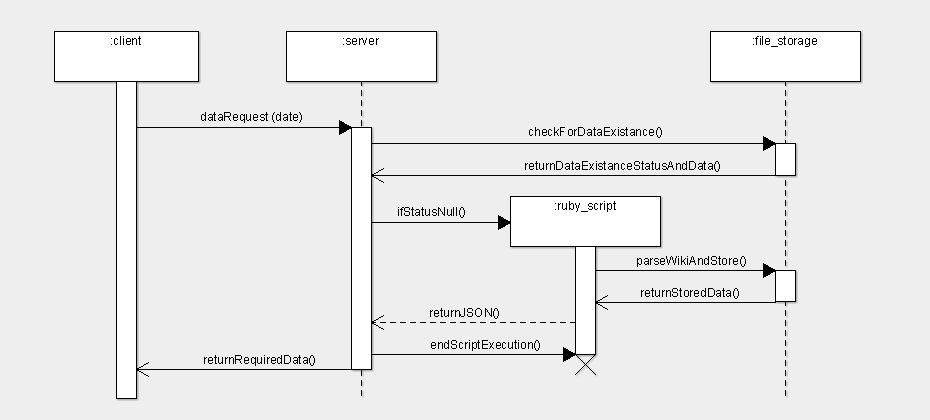
9.OS:Windows 8.1

1. **Design**

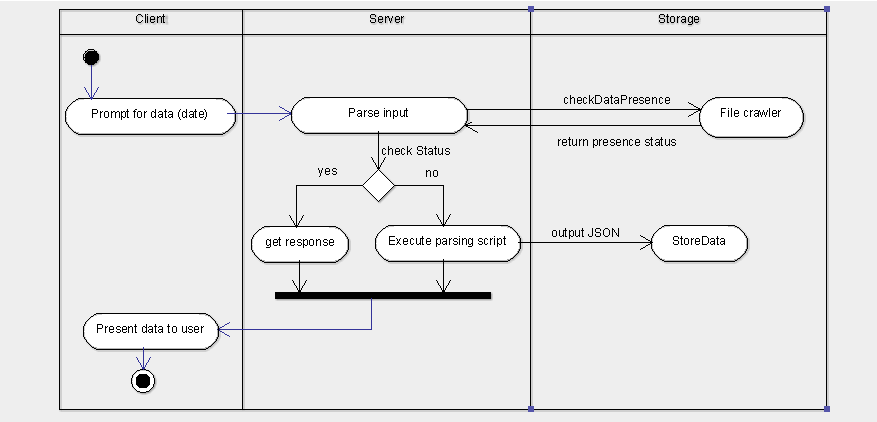
**USE-CASE DIAGRAM:**

****

**SEQUENCE DIAGRAM:**

****

**ACTIVITY DIAGRAM:**

****

1. **IMPLEMENTATION**

The application Pocket History has mainly 2 modules i.e. the client and the server module. The client module is written in JAVA over the Android SDK and the server is written in PHP over the Laravel framework.

There are mainly three stages of implementation such that the client receives the desired information. The stages are as follows:-

1. Sending the request from client to server.
2. Receive and process the request.
3. Responding with the result.
4. **Sending the request from server to client**

The request from the client is taken in the form of date in a specified format (dd/mm/yyyy), for example, 23/08/1994 viz. 23rd August 1994. Which means, the client expects a list of all the events, births and deaths (if any) that happened on 23rd August 1994. However, we provision two more types of inputs i.e. dd/mm or yyyy. This means we process client inputs for only day and month or only year as well.

In case A: When the input is of the format dd/mm (say 23/08), we provide a list of all the events, births and deaths that happened on 23rd august till the current year.

In case B: When the input is of the format yyyy (say 1994), we provide a list of all the events, births and deaths that happened in the year 1994.

As discussed before, all the client side code and validations are written in JAVA. The user interface of the application is written in XML using the tags provided by the Android SDK. After all the input validation, the request is made to the server.

1. **Receive and process the request**

The data receiving and processing is done by the server viz. hosted on Apache 2, running Laravel PHP MVC (Model View Control) framework. The server has a cron job scheduled for a script written in RUBY that makes a request to MediaWiki API and then parses the content to JSON format. The cron job runs on an interval of every 24 hours. All the requests are processed based on http web services. After receiving the response from the client the server performs a search on the file storage area to look for the particular information. The reason behind using the file storage is pretty simple, a huge amount of data. If the information is not existing, the code calls the RUBY script and gets the data, which is again parsed to JSON format and stored for further requests on the same information.

1. **Responding with the result**

As discussed earlier, all the processed data is stored in JSON format. Hence, the only thing required is to read the file and send the data to the client requesting the data. The clients receives the information where the JAVA code consumes the JSON and displays as required.