Ghulam Ishaq Khan Institute of Engineering Sciences &Technology

Faculty of Computer Science & Engineering

Project Document (Technical Manual)

Project Name: **QuantumPath**

Advanced Graph Algorithms for Optimized Campus Navigation

Group Members:

Ilsa Maryam (2023258)

Rabbin Batool (2023586)

Hiba Zulfiqar (2023246)

Course Details: DSA (CS-221)
Instructor: Sir Abdullah Bin Zarshaid
Submission Date: December 23, 2024

QuantumPath: Advanced Graph Algorithms for Optimized Campus Navigation

Abstract

QuantumPath is an advanced navigation system designed to simplify wayfinding across the Ghulam Ishaq Khan Institute (GIKI) campus. Leveraging advanced graph algorithms, real-world geospatial data, and AR/VR technology, the system offers efficient route calculations and immersive campus navigation. The integration of theoretical concepts, such as Dijkstra's and Bellman-Ford algorithms, with practical web development and visualization tools creates a user-friendly platform for students, staff, and visitors.

QuantumPath demonstrates the potential of technology in solving real-world problems while laying the groundwork for future enhancements such as mobile application development and multi-language support.

Introduction

Navigating large campuses like GIKI can be overwhelming, especially for new students and visitors. Traditional maps are often insufficient for real-time route optimization. QuantumPath addresses this problem by providing a robust, graph-based navigation system integrated with AR/VR capabilities.

This project underscores the importance of algorithmic problem-solving and modern visualization techniques in creating efficient, user-friendly systems. By combining C++ for algorithmic calculations with web technologies and AR/VR tools, QuantumPath ensures accurate, interactive, and immersive navigation.

Key Features:

- Real-time route optimization using Dijkstra's and Bellman-Ford algorithms.
- A user-friendly web interface for route selection and visualization.
- Immersive AR/VR navigation for a modernized user experience.
- Integration with Google Maps for enhanced geospatial accuracy.

System Requirements

Hardware

- A computer with at least 8 GB of RAM and a modern processor.
- Meta Quest Pro AR/VR headset for immersive navigation.

Software

- **Operating System**: Windows 10.
- **Programming Tools**:
    - C++ compiler (e.g., Code: Blocks, Visual Studio).
    - Visual Studio Code for web development.
- **Mapping Tools**: ArcMap 10.8 for geographical plotting.
- **AR/VR Tools: Lapentor for 360-degree visualization.**

Dependencies

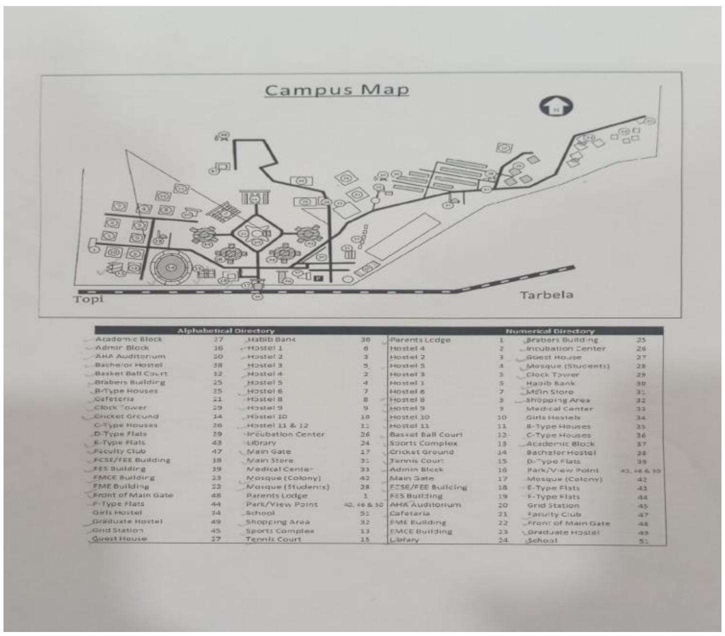
- Google Maps API for real-time location data.
- Web browser (Chrome).

System Design

Architectural Overview

The project is designed with modularity and scalability in mind. Key modules include:

1. **Graph Module**:
    - Represents the campus as a graph with nodes (locations) and edges (paths).
    - Calculates the shortest and most efficient routes using algorithms.



FID	Name	Longitude	Latitude
0	Main Gate	72.64441162	34.06817076
1	Clock Tower	72.64518665	34.06853819
2	Admin	72.64384096	34.06822645
3	AHA Auditorium	72.64423934	34.06974959
4	FCSE/ FEE	72.64348974	34.06952355
5	FCSE/ FEE	72.64305999	34.06876414
6	Tennis Court	72.64238185	34.06836558
7	Sports Ground	72.64152529	34.06894772
8	FES	72.64206225	34.06936555
9	FES	72.64262561	34.06970212
10	Academic Block	72.6430675	34.0702205
11	Academic Block	72.64316676	34.07053452
12	Library	72.6436385	34.07111185
13	FME	72.64460331	34.06955101
14	FMCE	72.6452253	34.07014423
15	Mosque	72.6414004	34.0694484
16	Guest House	72.645999	34.06869548
17	HBL Bank	72.64629011	34.06945418
18	Transport office	72.6471283	34.0694546
19	Tuc area	72.6471184	34.0700648
20	Tuc area	72.64723833	34.07076408
21	Cafe	72.64372605	34.06963965
22	Brabers Building	72.64495761	34.07150647
23	Incubation center	72.64575473	34.07166708
24	Graduate Hostel	72.64489218	34.07222031
25	New GH	72.64571309	34.07232143
26	New GH	72.64603432	34.07204423
27	Old GH	72.64636269	34.07176345
28	Old GH	72.6468481	34.07201091
29	Medical Center	72.64665298	34.0713875
30	Medical Center	72.64681478	34.07165876
31	Helipad	72.64406114	34.07268256
32	Faculty CLub	72.64268343	34.07262808
33	C-types Houses	72.64767139	34.07262006
34	Bechelor Hostel	72.6502388	34.0740146
35	D-types Houses	72.65119049	34.07375826
36	Sports Complex	72.6400504	34.0681848
37	BasketBall Court	72.6408476	34.068232
38	Hostel 9	72.64033551	34.0687106
39	Hostel 10	72.6395146	34.06868086
40	Hostel 1	72.64033551	34.06907942
41	Hostel 2	72.6395265	34.06897234
42	Hostel 3	72.64024033	34.0698349
43	Hostel 4	72.63936588	34.0698111
44	Hostel 5	72.64024033	34.07010854
45	Hostel 6	72.63936588	34.07009069
46	Hostel 8	72.64113625	34.0702228
47	Central Mess	72.64099581	34.06996577

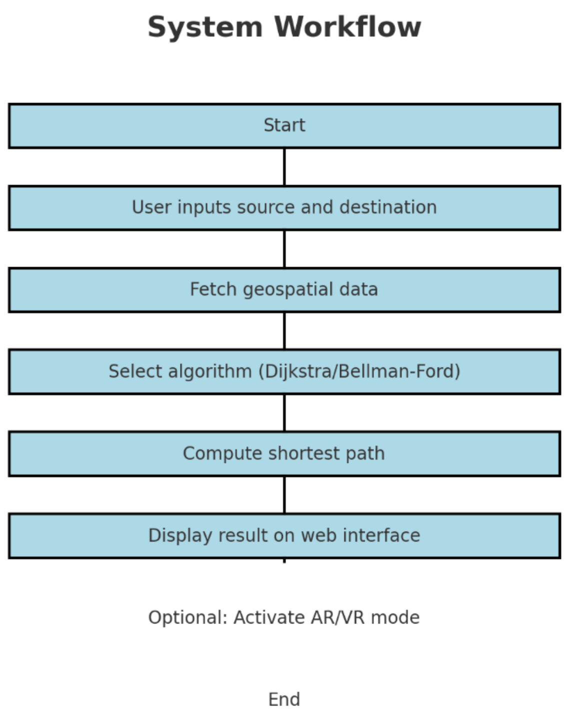
2. **Web Interface Module**:
    - Provides user interaction through a multi-page website.

- Features include route selection, map display, and AR/VR integration.

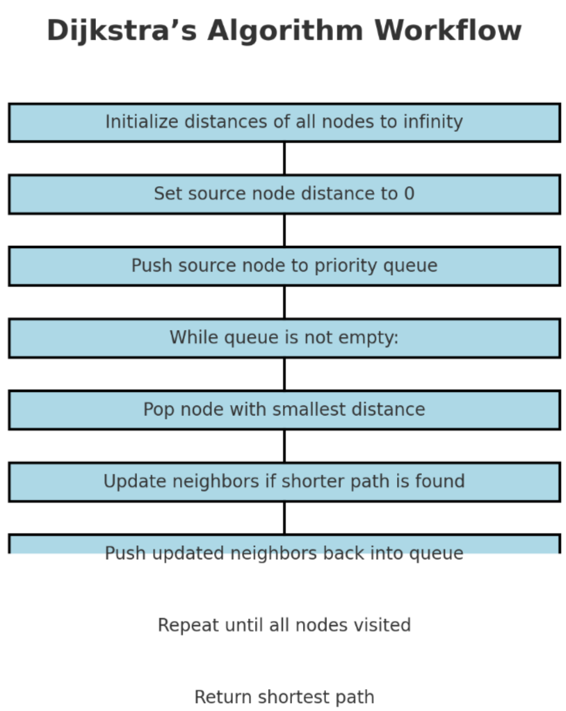
3. **AR/VR Module**:
   - Offers a 360-degree virtual tour of the campus.
   - Provides an immersive navigation experience using Meta Quest Pro.

Flowchart

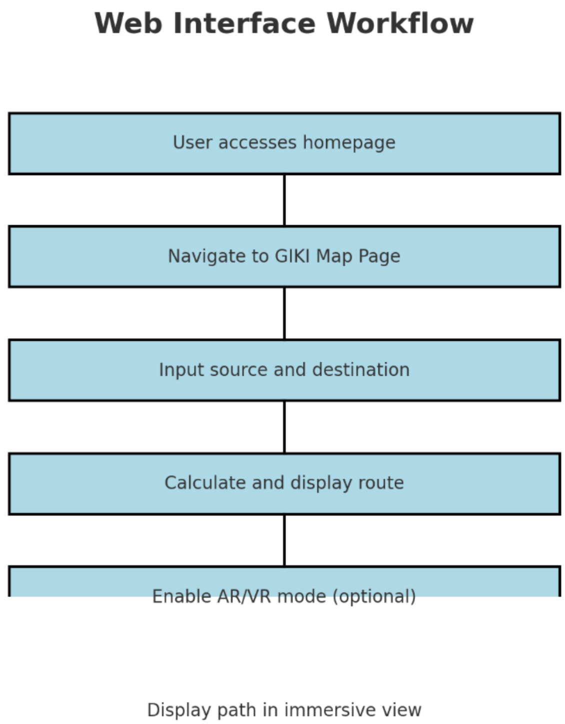
System Work Flowchart



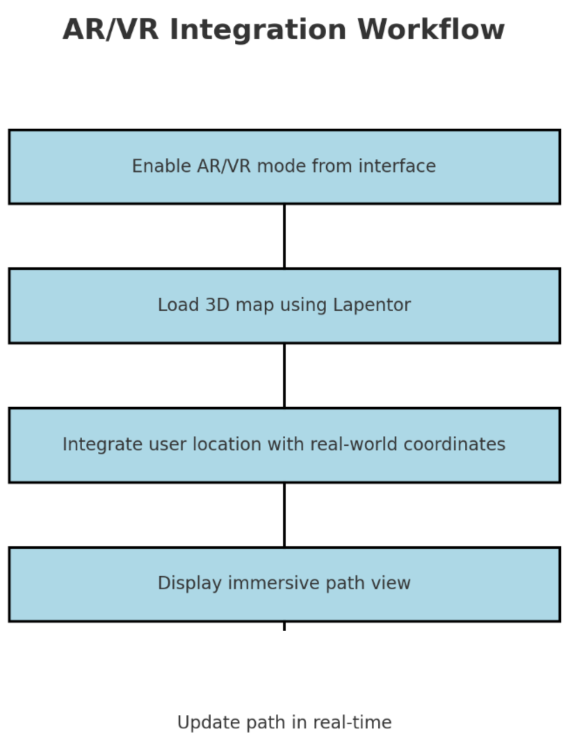
Dijkstra’s Work Flowchart



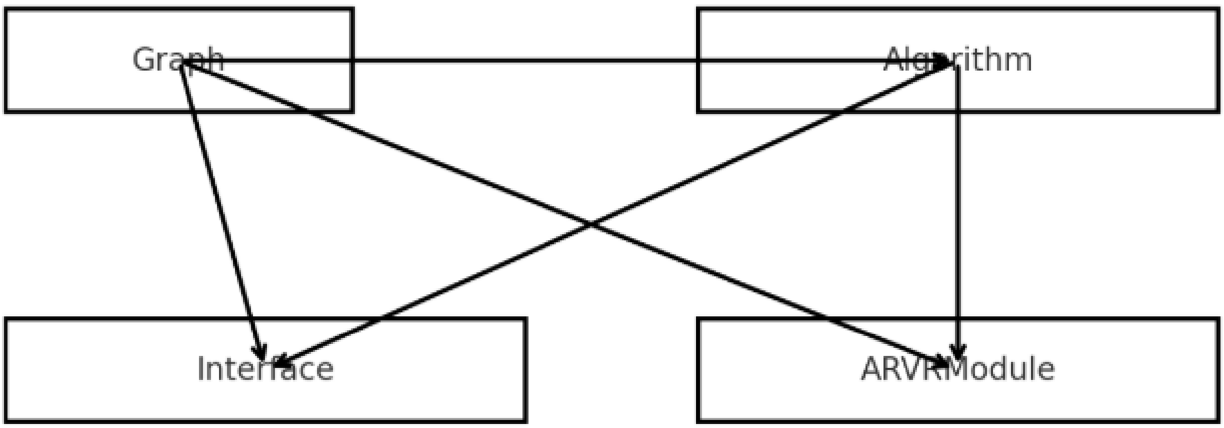
GUI Work Flowchart

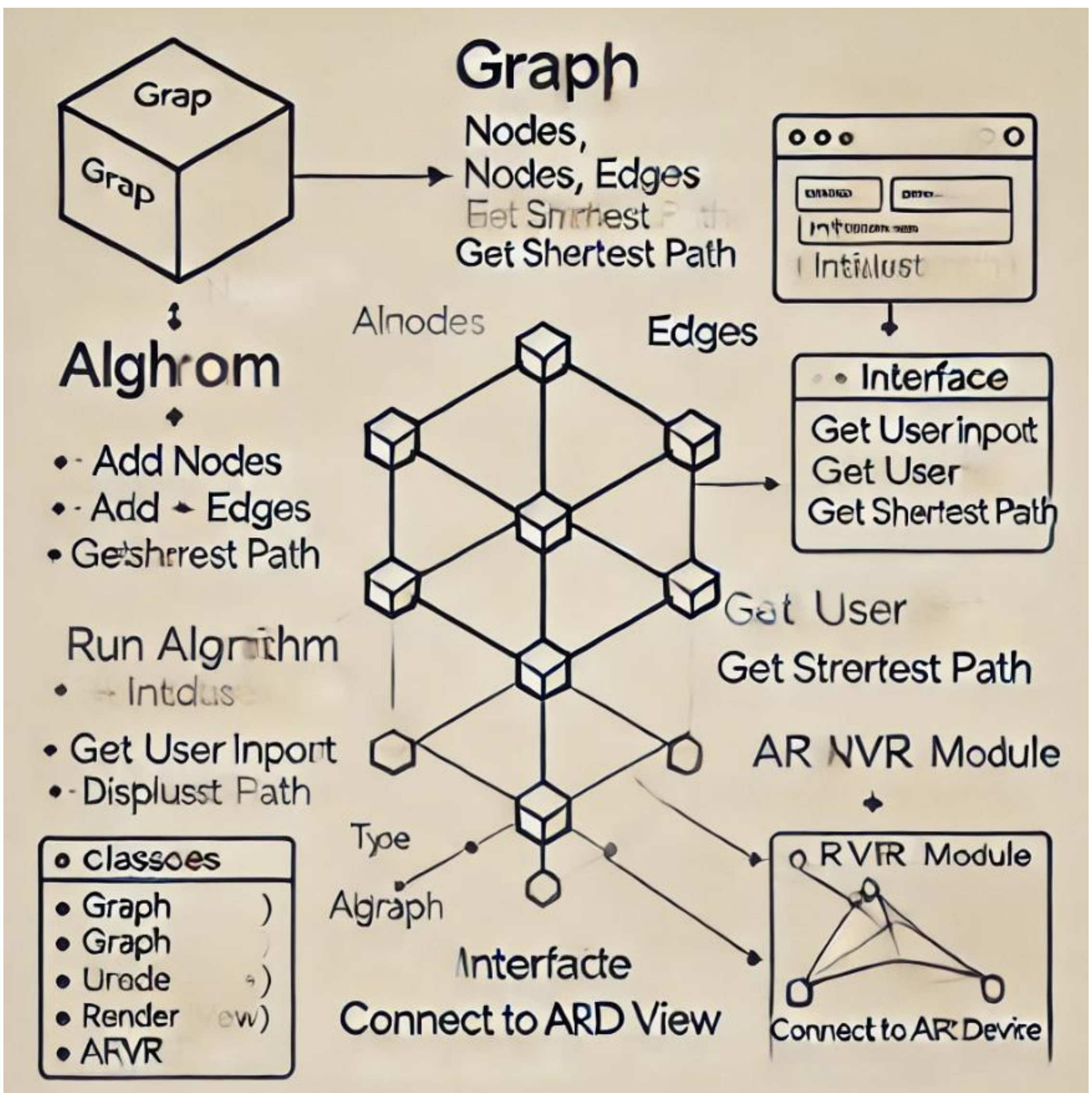


AR/VR Integration Work Flowchart

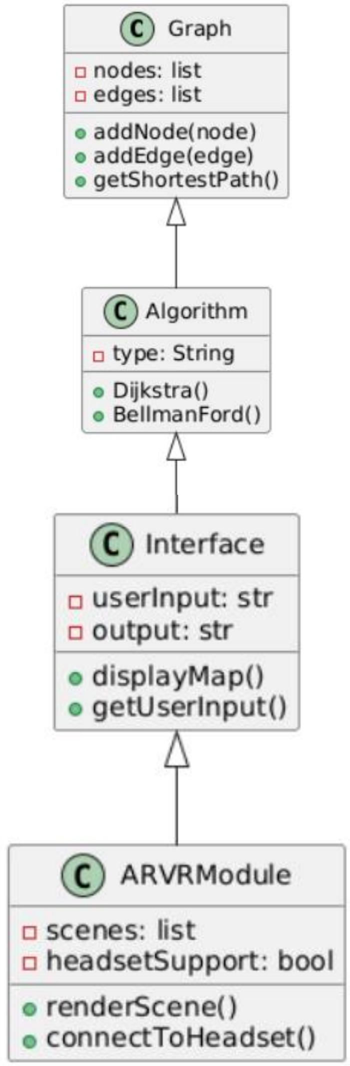


UML Diagram


Graph
Algorithm
Interface
ARVRModule


Graph
Grap
Grap
Nodes,
Nodes, Edges
Get Shertest Path
Alnodes
Edges
Alghrom
Add Nodes
Add Edges
Interface
Get User inpott
Get User
Get Shertest Path
Run Algrithm
Get User
Get Strertest Path
Get User Inport
AR NVR Module
Type
RVR Module
Classoes
Graph
Graph
Urede
Render
AFVR
Algraph
Interfacte
Connect to ARD View
Connect to AR Device

Proper UML Diagram Implementation



Implementation Details

Graph Representation

- Nodes represent campus locations with geographical coordinates.
- Edges represent paths, weighted by real-world distances.
- Data structure: adjacency list for efficient traversal.

Algorithms

1. **Dijkstra's Algorithm**: Finds the shortest path between nodes.
2. **Bellman-Ford Algorithm**: Handles graphs with negative weights, though not present in the current dataset.

Tools and Frameworks

- **Frontend**: HTML, CSS, JavaScript for dynamic web interfaces.
- **Backend**: C++ for graph and algorithmic computations.
- **Mapping**: ArcMap 10.8 for precise road plotting.
- **Visualization**: Lapentor for AR/VR integration.

Testing and Results

Testing Overview

- **Unit Testing**:
  Verified individual components, such as the shortest path calculation and edge weight validation.
- **Integration Testing**:
  Ensured seamless interaction between the graph module, web interface, and AR/VR visualization.
- **System Testing**:
  Simulated user scenarios to validate end-to-end functionality.

Test Cases

Test Case ID	Test Scenario	Input	Expected Output	Actual Output	Status
TC-01	Calculate shortest path between two valid nodes	Source: A, Destination: B	Display shortest path with distance and route visualization	Correct path displayed with distance and visualization	Passed
TC-02	Handle invalid source or destination input	Source: Invalid, Destination: C	Display error message: "Invalid input. Please try again."	Error message displayed	Passed
TC-03	Display AR/VR view	AR/VR button clicked	Launch 360-degree campus view in immersive mode	AR/VR campus view launched successfully	Passed
TC-04	Validate negative weights in graph	Graph with negative edge weights	Bellman-Ford handles negative weights correctly	Negative weights handled without issues	Passed
TC-05	Test graph connectivity	Run traversal the nodes	All nodes in the graph are connected	Graph traversal confirmed connectivity	Passed

TC-06	Web Interface responsiveness	Open GUI on Chrome	Interface adjusts to screen size	Responsive GUI design works as expected	Passed
TC-07	Edge case: Same source and destination	Source: A, Destination: A	Display "Source and destination are the same"	Message displayed correctly	Passed
TC-08	Test performance on large input graph	Graph with nodes and edges	Shortest path computed within acceptable time.	Performance within acceptable range	Passed
TC-09	Validate real-time Google Maps integration	Input any two locations	Correct route displayed using Google Maps API	Route displayed accurately	Passed
TC-010	Test error handling during API failure	Disconnect from the internet	Display error message: "Unable to fetch map data."	Correct error message displayed	Passed

Results

- Achieved precise shortest path calculations.
- Integrated a fully functional web interface.
- Successfully rendered AR/VR campus views.

User Guide

Installation

1. Clone the project repository from GitHub.
2. Install necessary dependencies (Google Maps API, Lapentor setup).

Configuration

- Configure the Google Maps API key in the project settings.
- Set up Meta Quest Pro for AR/VR navigation.

Execution

- Run the backend C++ module to load graph data.
- Open the web interface in a browser.

- Select source and destination points to calculate the shortest path.
- Access AR/VR view for an immersive experience.

Future Enhancements

- **Mobile Application**: Develop an Android/iOS app for on-the-go navigation.
- **Enhanced AR Features**: Include real-time path overlay in AR view.
- **Real-Time Traffic**: Incorporate live traffic data for route adjustments.
- **Accessibility**: Add voice navigation and wheelchair-accessible paths.
- **Multi-Language Support**: Provide support for multiple languages.

Conclusion

QuantumPath effectively combines advanced graph algorithms with cutting-edge technologies to solve real-world navigation challenges. The system's modular design and use of AR/VR offer a modernized, interactive experience. This project not only achieves its objectives but also sets the stage for future innovations in campus navigation.

References

- **Algorithm Resources**:
  1. GeeksforGeeks - Dijkstra's Algorithm
  2. GeeksforGeeks - Bellman-Ford Algorithm
  3. Programiz - Graph Data Structure
- **C++ Programming**:
  1. The Cherno - C++ Programming Tutorials (YouTube)
  2. CodeBeauty - C++ Projects and Tips (YouTube)
  3. cplusplus.com - C++ Standard Library Reference
- **Mapping and AR/VR**:
  1. Lapentor Official Documentation
  2. Meta Quest Pro Guide
- **Educational Resources**:
  1. Khan Academy - Algorithms
  2. MIT OpenCourseWare - Introduction to Algorithms