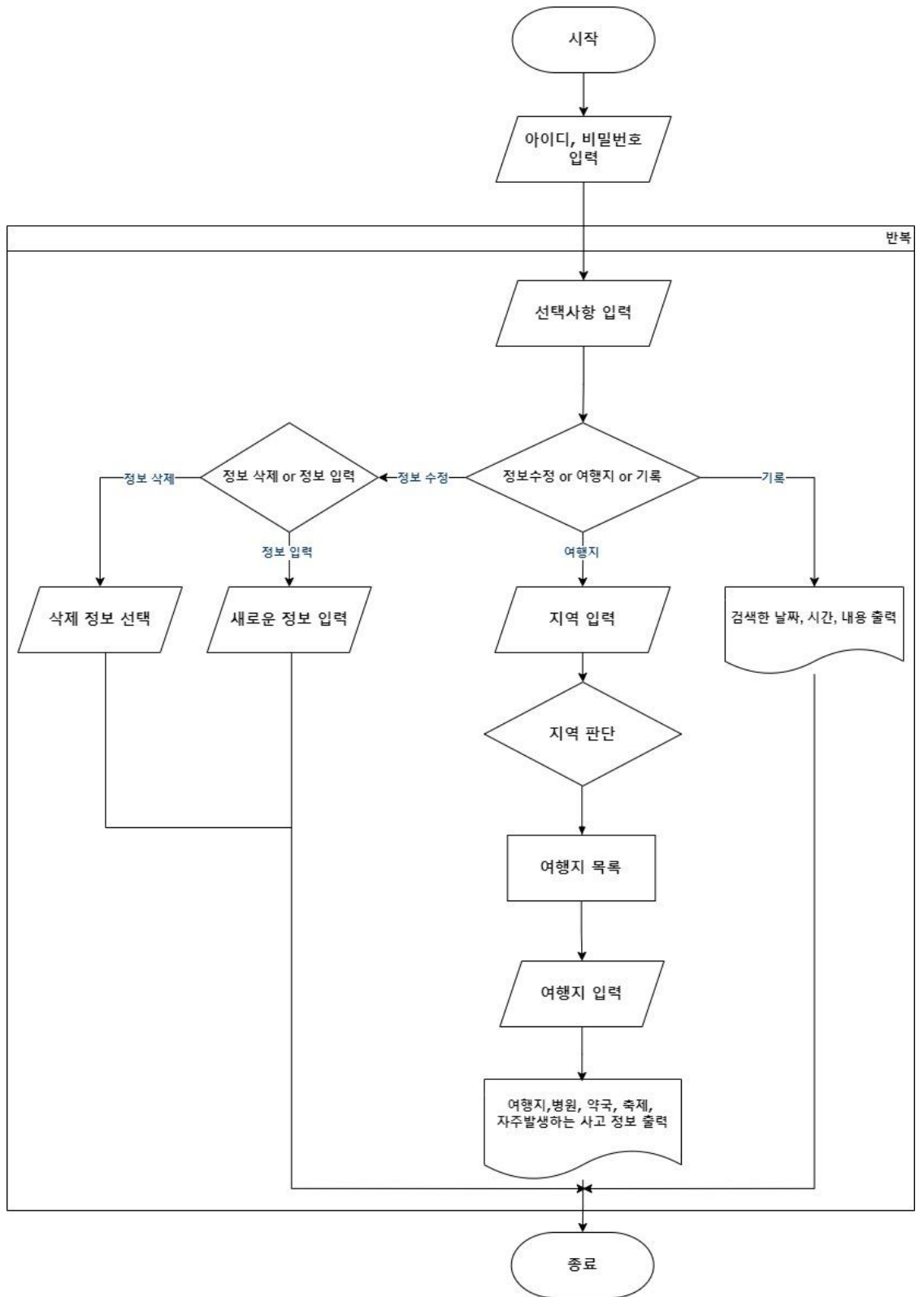


개발완료 보고서

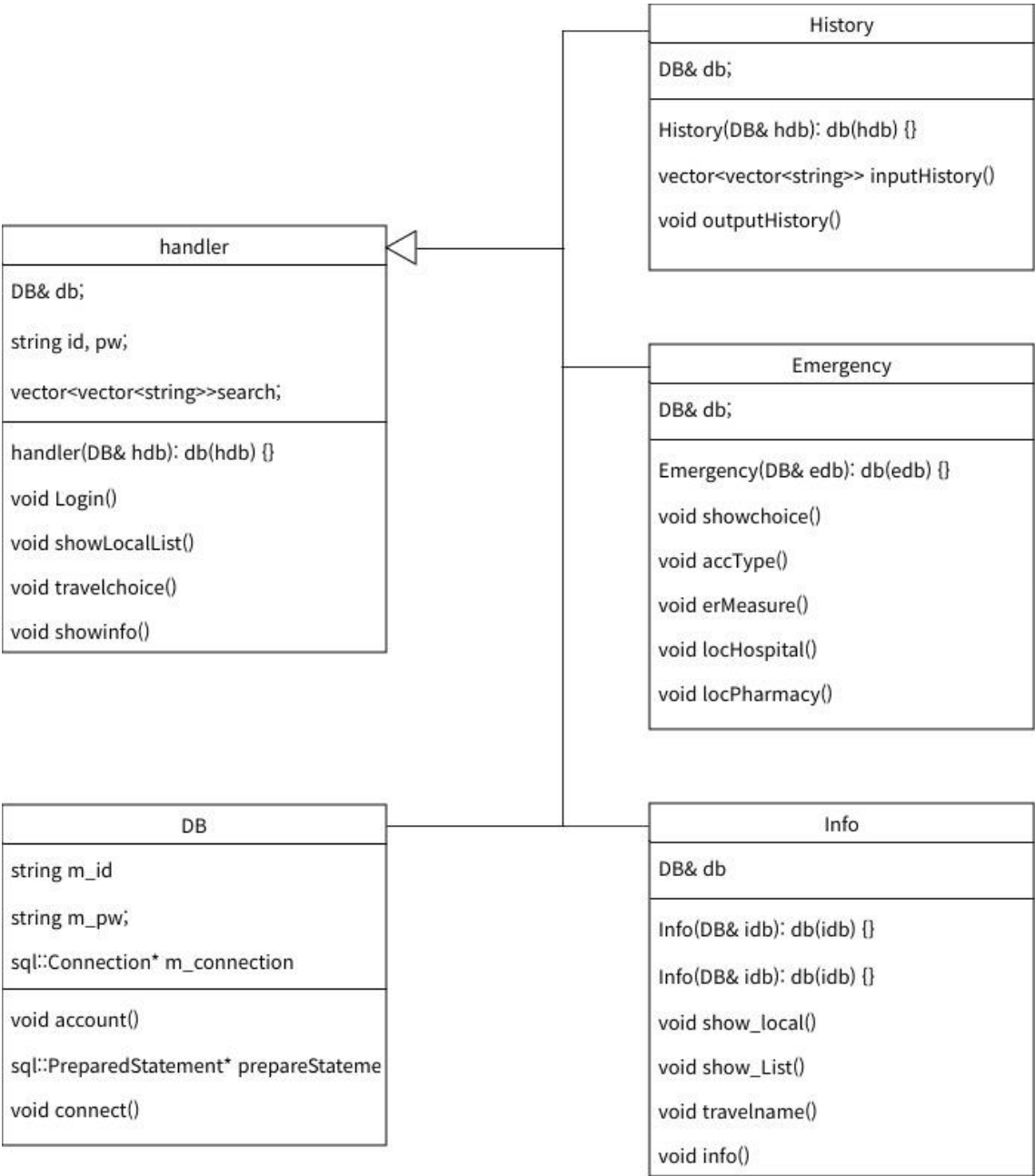
제출일 : 2024년 5월 10일

| | | | | | | | | | | | |
|-----------|---|---------------|-------|--|--------------------------|--|-------|--|-------|--|--|
| | 참여인원 | | | | | | | | | | |
| 참여자 | 유 시 온 | | | | | | | | | | |
| 개발프로젝트 소개 | | | | | | | | | | | |
| 프로젝트 명 | “또디비자” “AGAIN DBZA” | | | | | | | | | | |
| 활동일시 | 5 / 7 ~ 5 / 10 | | 장소 | | 광주인력개발원 공학1관 2층 드론융합실 | | | | | | |
| 주요주제 | DB와 VS코드를 연동해서 지역별 관광지 및 사고예방에 대한 정보 조회 및 출력 | | | | | | | | | | |
| 개발목적 | DB를 통해 생성한 테이블 파일 저장, DB를 VS코드에 연동한 데이터를 바탕으로 C++언어를 이용한 프로그램 개발 | | | | | | | | | | |
| 개발환경 | Ubuntu 16.04 Linux os / Visual Studio Code 1.88.1 / g++ 13.2.0 / gnu++17 / mariaDB | | | | | | | | | | |
| 일 정 표 | 항 목 | | 05/07 | | 05/08 | | 05/09 | | 05/10 | | |
| | 개발계획서 및 일정표작성 | | | | | | | | | | |
| | 구현 | 데이터 분석 및 전처리 | | | | | | | | | |
| | | DB 생성 및 연동 | | | | | | | | | |
| | | 클래스 분류 | | | | | | | | | |
| | | 히스토리 생성 | | | | | | | | | |
| | | 최종 점검 | | | | | | | | | |
| | 제출 | 완료보고서 작성 및 제출 | | | | | | | | | |

순서도



다이어그램



요구 분석서

| 분 류 | 요구 분석 내용 | |
|------------------|----------------|---|
| | 발생 조건 세부 분류 | 세 부 내 용 |
| 시작 | 로그인 | mariaDB 아이디, 패스워드 입력 받기 |
| 메뉴 | 선택 사항 입력 | 1) 여행지 정보 2) 기록 3) 정보 입력 4) 정보 삭제 |
| 여행지 | 지역 입력 | 검색 원하는 지역 입력 |
| | 관광지 목록 | 해당 관광지 목록 출력 |
| | 정보 입력 | 검색 원하는 관광지 번호 입력 |
| | 정보 출력 | - 관광지 명, 연락처, 주소 - 축제(축제기간) - 자주 발생하는 사고 정보 - 병원 정보(주소, 연락처) - 약국 정보(주소, 연락처) |
| 응급 의료시설 정보 | 응급상황 대처방안 | 응급상황 유형별 목록 출력 후 사용자가 유형 입력시 대처방안 사이트 출력 |
| | 지역별 병원 | 사용자가 지역 입력시 해당 지역에 대한 병원 정보 출력 |
| | 지역별 약국 | 사용자가 지역 입력시 해당 지역에 대한 약국 정보 출력 |
| 기 록 | 기록 리스트 | - 현재 까지 검색한 기록 리스트 출력 (검색한 날짜, 시간, 내용) - 최대 30개 까지 기록 가능 |
| | 기록 재검색 | -기록 리스트 중 재검색 원하는 번호 입력 시 정보 다시 출력 |

| | | |
|-------|--|--------------------|
| 클래스 | class DB | mariadb 데이터 연동 클래스 |
| | class Emergency | 응급상황, 병원, 약국 클래스 |
| | class History | 검색 기록 클래스 |
| | class Info | 여행지 정보 클래스 |
| | class handler | 핸들러 클래스 |
| 멤버 변수 | DB& db | DB클래스 참조 변수 |
| | string id, pw | mariadb 아이디,패스워드 |
| | vector<vector<string>>>search | 검색기록 벡터 변수 |
| 멤버 함수 | void account | mariadb로그인 |
| | sql::PreparedStatement* prepareStatement | 쿼리문 객체 생성 |
| | void connect | db 연결 |
| | void show_local | 지역 목록 출력 |
| | void show_List | 여행지 목록 출력 |
| | void travelname | 여행지 이름 출력 |
| | void info | 여행지 정보 검색 |
| | void showErMedi | 선택사항 항목 출력 |
| | void accType | 응급상황 유형 출력 |
| | void erMeasure | 응급상황 대처방안 검색 |
| | void locHospital | 지역별 병원 검색 |
| | void locPharmacy | 지역별 약국 검색 |
| | vector<vector<string>>> inputHistory | 검색 기록 입력 |
| | void outputHistory | 검색 기록 출력 |
| | void Login | 아이디, 패스워드 입력 |
| | void showchoice | 선택사항 항목 출력 |
| | void chTravelHistory | 여행지정보, 기록정보 선택 |
| | void Search | 메인 검색 |

코드 설명

```
class DB : mariaDB 데이터베이스 연동 클래스
void account(const string& id, const string& pw)
    : 입력받은 아이디와 패스워드를 함수를 이용해 클래스의 멤버변수의 값을 변경시킨다.
sql::PreparedStatement* prepareStatement(const string& query)
    : 매개변수를 쿼리문 객체로 생성한다.
void connect()
    : 아이피 주소와 아이디, 패스워드등 이용해 mariaDB에 접속한다.
```

```
class Info : 여행지 정보 및 근처 약국, 병원 축제 정보 클래스
void show_local()
    : db 쿼리문을 이용해 중복없이 지역 리스트 출력한다.
void show_List(string loc)
    : 지역을 입력받아 해당 지역의 여행지 리스트를 출력한다.
void travelname(int code)
    : 여행지 코드를 입력하면 여행지 이름만 출력한다.(검색 기록 위해 필요)
void info(int code)
    : 여행지 코드 입력시 해당 여행지 근처 축제, 병원, 약국, 자주 발생 사고유형
    정보를 출력한다.
```

```
class Emergency : 응급상황, 지역별 병원, 지역별 약국 정보 클래스
void showErMEdi()
    : 사용자가 선택할 때 응급상황, 지역별 병원, 지역별 약국 보기 리스트 출력한다.
void accType()
    : db 쿼리문을 이용해 응급상황 유형별 리스트를 출력한다.
void erMeasure(string type)
    : db 쿼리문을 이용해 입력받은 응급상황에 따른 해당 조치방안을 출력한다.
void locHospital(string loc)
    : db 쿼리문을 이용해 입력받은 해당 지역의 병원이름, 주소, 연락처를 출력한다.
void locPharmacy(string loc)
    : db 쿼리문을 이용해 입력받은 해당 지역의 약국 이름, 주소, 연락처를 출력한다.
```

```
class History : 여행지 정보 검색 기록 저장 클래스
vector<vector<string>> inputHistory(string loc, int code)
    : 여행지 정보에서 검색한 지역, 여행지 코드를 매개변수로 받아
    2차원 벡터에 저장한다. (벡터사이즈가 30이 넘으면 맨 앞을 지우고 저장)
void outputHistory(vector<vector<string>> history, Info& idb)
    : 검색한 기록을 반복문을 사용하여 출력한다. Info 클래스의 함수를 이용하여
    재검색이 가능하도록 하였다.
```

```
class handler : 핸들러 클래스
void Login()
    : mariaDB 아이디와 패스워드를 입력받아 DB 접속을 진행한다.
void showchoice()
    : 프로그램 실행시 선택항목 보기 리스트를 출력한다.
void chTravelHistory()
    : (여행지 정보 검색, 검색 기록) 선택 항목 보기를 출력한다.
void Search()
    : 사용자에게 선택항목을 입력 받으면 정보를 볼수 있도록 각 클래스의
    선택항목 출력 함수, 정보 출력 함수들을 실행한다.
```

후 기

이번 프로젝트는 계획했던 것과 다르게 많이 바뀌게 된 프로젝트였습니다. 프로젝트 조건을 보고 쿼리문을 이용하여 테이블에 값을 추가하는 것과 값을 제거하는 것을 계획하였습니다.

진행하다 보니 테이블들이 기본키, 외래키로 관계가 형성되어 있다 보니 값을 새로 추가하거나 삭제했을 시 문제가 발생했습니다. 쉽게 생각했던 것과 다르게 테이블을 생성할 때 제약조건 따로 주거나 외래키로 관계가 형성되어 있으면 1개의 테이블만 아니라 관계된 테이블도 같이 봐야 했습니다.

이번 개인 프로젝트를 진행하면서 DB에 대해 놓쳤던 부분을 알아가게 되었지만, C++을 대해서는 여전히 사용할 수 있는 부분만 사용하고 있다는 느낌을 받았던 프로젝트였던 것 같습니다. 어렵다고 느꼈던 부분에 대해 포기하지 말고 좀 더 찾아보고 공부해서 다음에는 더 넓게 활용하여 프로젝트를 진행하고 싶습니다.