

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Кафедра ЕОМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Мобільний застосунок «Путівник по місту
Харків»»

виконав:
здобувач гр. КІУКІу-22-1
Стоянов Д.О.

керівник:
доц. каф. ЕОМ
Філімончук Т.В.

Харків 2025

2

Мета роботи

Метою кваліфікаційної роботи є створення мобільного застосунку-путівника для міста Харків, який забезпечує зручний доступ до туристичної та довідкової інформації, дозволяє орієнтуватися на мапі, будувати маршрути до визначних місць та взаємодіяти з об'єктами інфраструктури.

Задачі кваліфікаційної роботи:

- провести аналіз предметної області;
- оцінити наявні аналоги;
- встановити перелік основних функцій та обмежень застосунку;
- обрати технології для реалізації застосунку;
- розробити структуру застосунку;
- реалізувати інтерфейс користувача;
- провести тестування застосунку.

Актуальність обраної теми

Мобільні застосунки відіграють дедалі важливішу роль у сфері навігації та туристичного супроводу, що робить розробку мобільного путівника для міста Харків актуальним та своєчасним рішенням.

Актуальність теми підтверджується такими факторами:

- активний розвиток мобільних технологій та попит на цифрові сервіси;
- мала кількість локалізованих туристичних сервісів для українських міст;
- туристичний потенціал Харкова залишається недостатньо реалізованим;
- існуючі застосунки не враховують місцевих особливостей;
- потреба в офлайн-доступі до перевіреної інформації про об'єкти, маршрути й події.

Існуючі застосунки-путівники

○ Kiev



- + мінімалістичний інтерфейс
- непрацюючі розділи
- застаріла карта

○ LvivGuide



- + яскраве візуальне оформлення
- зависання та помилки при завантаженні карти
- застаріла карта

○ Kolomyia guide and routes



- + зручне розділення контенту за вкладками
- відсутній опис місць
- обмежений функціонал

○ Львівщина



- + чітка структура та описи об'єктів
- відсутність фільтрів на мапі та маршрутизації

Технології розробки

5

- Dart
- Flutter
- SQLite
- OpenStreetMap

- Flutter_bloc, bloc
- Url_launcher
- Http

- Flutter_map
- Latlong2
- OSRM
- Android_intent_plus
- Geolocator



SQLite



Архітектура застосунку

6

Основні модулі:

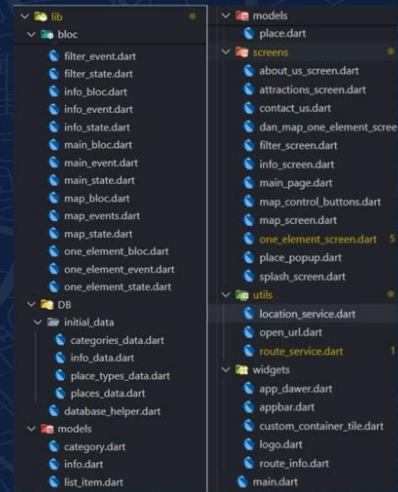
- **UI** (screens + widgets): екрани, віджети, навігація;
- **BLoC** (bloc/*.dart): обробка подій, оновлення станів, логіка взаємодії;
- **Моделі** (models/*.dart): структури даних, які циркулюють між модулями;
- **Сервіси** (utils/*.dart): сервіси геолокації, маршрутів та відкриття посилань;
- **База даних** (DB + initial_data): локальна база даних.



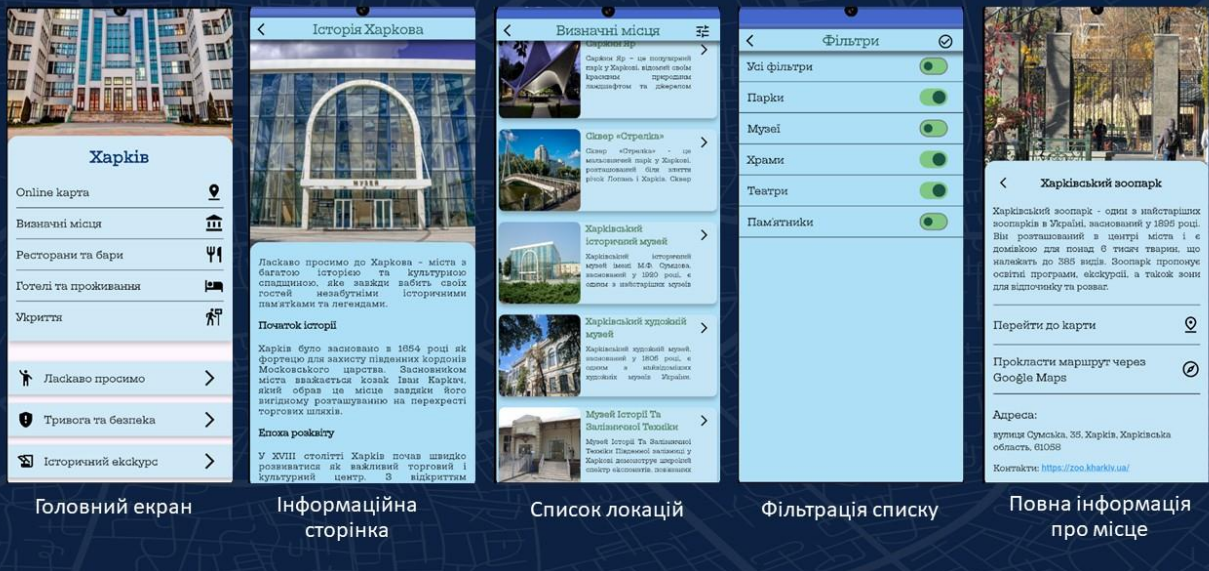
Структура застосунку

Основні каталоги та файли:

- **bloc** – логіка керування станом (файли типу bloc, event, state для кожного модуля);
- **DB** – взаємодія з локальною базою даних (database_helper.dart, initial_data/*.dart);
- **models** – Dart-класи для збереження структури об'єктів (місця, категорії, опції тощо);
- **screens** – екрани інтерфейсу користувача (головна, карта, фільтри, контактна інформація тощо);
- **utils** – допоміжні сервіси (геолокація, маршрути, посилання);
- **widgets** – багаторазові віджети інтерфейсу (AppBar, Drawer, Route_info тощо);
- **main.dart** – точка входу в застосунок, ініціалізує навігацію та стан.

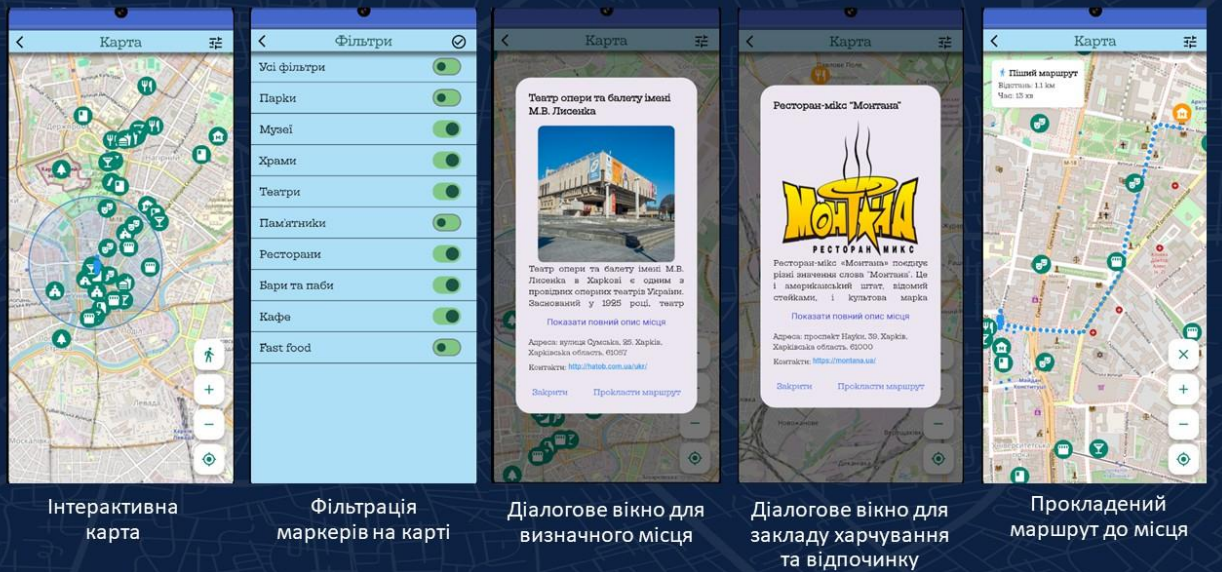


Екрани мобільного застосунку



Екрани мобільного застосунку

9



Висновки

10

У ході виконання кваліфікаційної роботи було виконано всі поставлені задачі, а саме:

- проведено аналіз існуючих рішень;
- проаналізовано сучасні технології розробки мобільних застосунків;
- вивчено відповідну літературу;
- розроблено архітектуру застосунку;
- реалізовано базу даних для збереження інформації;
- створено повнофункціональний мобільний застосунок;
- протестовано основні функції.

Розроблений застосунок має інтуїтивно зрозумілий інтерфейс, зручну навігацію містом та ефективну структуру, що забезпечує стабільну роботу, простоту підтримки та можливість подальшого розвитку.

ДОДАТОК Б

Код програми

Б.1 Клас `_MapScreenState`

```
class _MapScreenState extends State<MapScreen> with
WidgetsBindingObserver {Map<int, bool> selectedFilters = {};
  List<int> allowedTypeIds = [];late final MapController
_mapController;late final LocationService _locationService;
  bool _isMapReady = false;LatLng? _userLocation;bool
_isLoadingLocation = false;bool _isTrackingEnabled = true;
  StreamSubscription<LatLng>? _locationSubscription;List<LatLng>
_routePoints = [];bool _isLoadingRoute = false;bool _showRoute =
false;double? _routeDistance;double? _routeDuration;ListItem?
_selectedDestination;@override void initState()
{super.initState();WidgetsBinding.instance.addObserver(this);_ma
pController = MapController();_locationService =
LocationService();BlocProvider.of<MapBloc>(context).add(FetchMap
Items(allowedTypeIds));_initializeLocation();}@overridevoid
dispose()
{WidgetsBinding.instance.removeObserver(this);_locationSubscript
ion?.cancel();_locationService.dispose();super.dispose();}void
_initializeLocation()
{_getCurrentLocation();_locationSubscription =
_locationService.locationStream.listen((location) {if (mounted)
setState(() => _userLocation = location);});if
(_isTrackingEnabled)
_locationService.startLocationTracking();}Future<void>
_getCurrentLocation() async {if (_isLoadingLocation)
return;setState(() => _isLoadingLocation = true);final location
= await _locationService.getCurrentLocation();if (mounted)
setState(() {_userLocation = location;_isLoadingLocation =
false;});} Future<void> _buildWalkingRoute() async { if
(_selectedDestination == null) return;if (_userLocation == null)
await _getCurrentLocation();if (_userLocation == null) return;
  setState(() => _isLoadingRoute = true); try {final routeInfo
= await RouteService.buildWalkingRoute( from: _userLocation!,to:
LatLng(_selectedDestination!.latitude,_selectedDestination!.long
itude),);if (routeInfo != null && mounted) { setState(()
{_routePoints = routeInfo.points;_routeDistance =
routeInfo.distance;_routeDuration =
routeInfo.duration;_showRoute = true;_isLoadingRoute =
false;});_fitRouteInView();}} catch (_) {if (mounted)
setState(() => _isLoadingRoute = false);}}void _fitRouteInView()
{if (_routePoints.isEmpty || !_isMapReady) return;double minLat
= _routePoints.first.latitude, maxLat = minLat;double minLng =
_routePoints.first.longitude, maxLng = minLng;for (final p in
_routePoints) {minLat = minLat < p.latitude ? minLat :
p.latitude;maxLat = maxLat > p.latitude ? maxLat :
```

```

p.latitude;minLng = minLng < p.longitude ? minLng :
p.longitude;maxLng = maxLng > p.longitude ? maxLng :
p.longitude;}_mapController.fitCamera(CameraFit.bounds(bounds:
LatLngBounds(LatLng(minLat - 0.001, minLng -
0.001),LatLng(maxLat + 0.001, maxLng + 0.001),),));}
Widget _buildMap(MapLoaded state) {
  final markers = <Marker>[];for (final place in state.places)
{markers.add(Marker(width: 35,height: 35,
  point: LatLng(place.latitude, place.longitude),
  child: GestureDetector(
    onTap: () => setState(() => _selectedDestination =
place), child: Container(
  decoration: BoxDecoration(
    color: _selectedDestination?.id == place.id ?
Colors.orange : Color(0xFF0D836F),
    shape: BoxShape.circle,),
    child: Icon(getIconPath(place.placeTypeId) as
IconData,color: Colors.white, size: 24),),),
  ));}
  if (_userLocation != null) {
    markers.add(Marker(
      width: 50,
      height: 50,
      point: _userLocation!,
      child: Icon(Icons.boy_sharp, color: Colors.blue, size:
56),));}
  return FlutterMap(
    mapController: _mapController,
    options: MapOptions(
      initialCenter: _userLocation ?? LatLng(49.9935,
36.2304),initialZoom: 13.0,minZoom: 6.0,maxZoom: 18.0,
      onMapReady: () => setState(() => _isMapReady = true),
    ),children: [TileLayer(urlTemplate:
'https://tile.openstreetmap.org/{z}/{x}/{y}.png',userAgentPackag
eName: 'com.example.app',),
    if (_userLocation != null && !_showRoute)
      CircleLayer(circles: [CircleMarker(
        point: _userLocation!,radius: 750,color:
Color(0xFF1976D2).withOpacity(0.13),
        borderColor: Color(0xFF1976D2).withOpacity(0.7),
        borderStrokeWidth: 2.5,
        useRadiusInMeter: true,)],),if (_showRoute &&
_routePoints.isNotEmpty)PolylineLayer(
      polylines: [Polyline(points: _routePoints,color:
Colors.blue,strokeWidth: 8.0,pattern:
StrokePattern.dotted(),),),),MarkerLayer(markers:
markers),],);}

```

Б.2 Клас PlacePopupWithRoute

```

class PlacePopupWithRoute {static Future<void> showPlaceDialog(
  BuildContext context,dynamic place, {VoidCallback?
onBuildRoute,VoidCallback? selectedDestinationOff,

```

```

    }) async {
      final result = await showDialog<String>(
        context: context,
        barrierDismissible: true,
        builder: (context) => AlertDialog(
          title: Text(place.title, style: TextStyle(fontSize: 18,
fontWeight: FontWeight.bold)),
          content: Column(mainAxisSize: MainAxisSize.min,
crossAxisAlignment: CrossAxisAlignment.start,
children: [Center(child: ClipRRect(
borderRadius: BorderRadius.circular(10),
child: Image.network(place.imageUrl,
errorBuilder: (_, __, ___) =>
Image.asset('assets/Logo2.png', width: 250, height: 250),
width: 250,height: 250,
fit: BoxFit.fill,)),),),
Text(place.description, maxLines: 4, overflow:
TextOverflow.clip),
Center(
child: TextButton(
onPressed: () => Navigator.push(
context,
MaterialPageRoute(builder: (context) =>
OneElementScreen(item: place)),
),
child: Text('Показати повний опис місця'),
),
),
if (place.address?.isNotEmpty ?? false)
Text('Адреса: ${place.address}', style:
TextStyle(fontSize: 13)),
if (place.contactInfo?.isNotEmpty ?? false)
Row(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text('Контакти: ', style: TextStyle(fontSize:
13)), Expanded(child:
UrlLauncherHelper.buildClickableText(context,place.contactInfo!,
defaultStyle: TextStyle(fontSize: 13),linkStyle:
TextStyle(fontSize: 13, color: Colors.blue, decoration:
TextDecoration.underline),),),],),),
actionsPadding: EdgeInsets.fromLTRB(30, 0, 30, 30),
actions: [Row(mainAxisAlignment:
MainAxisAlignment.start,children: [TextButton(onPressed: () =>
Navigator.of(context).pop('closed'),child:
Text('Закрити'),),Spacer(),TextButton(onPressed: () =>
Navigator.of(context).pop('route'),child: Text('Прокласти
маршрут'),),),],),),]);
      if (result == 'closed' && selectedDestinationOff != null) {
        selectedDestinationOff();
      } else if (result == 'route' && onBuildRoute != null) {
        onBuildRoute();
      }
    }
  }
}

```

Б.3 Клас RouteService

```

class RouteService {static const String _osrmBaseUrl =
'https://router.project-osrm.org/route/v1';
  static const Duration _requestTimeout = Duration(seconds: 30);
  static const int _maxRetries = 3;
  static Future<RouteInfo?> buildWalkingRoute({
    required LatLng from, required LatLng to,}) async {
    if (!_isValidCoordinate(from) || !_isValidCoordinate(to)) {
      throw Exception('Некоректні координати');}
final profiles = ['foot', 'walking'];Exception? lastException;
  for (final profile in profiles) {for (int attempt = 0;
attempt < _maxRetries; attempt++) {try {final result = await
_attemptRouteRequest(from, to, profile);if (result != null)
return result;} on TimeoutException {lastException =
Exception('Перевищено час очікування (спроба ${attempt +
1}'));await Future.delayed(Duration(milliseconds: 1000 *
(attempt + 1)));} catch (e) {lastException = Exception('Помилка:
$e');await Future.delayed(Duration(milliseconds: 500));
}}}throw lastException ?? Exception('Не вдалося
побудувати маршрут');}
  static Future<RouteInfo?> _attemptRouteRequest(LatLng from,
LatLng to, String profile) async {final url =
_buildRouteUrl(from, to, profile);final response = await
http.get(Uri.parse(url), headers: {'User-Agent': 'Flutter-
App/1.0', 'Accept': 'application/json',}).timeout(_requestTimeout)
;if (response.statusCode == 200) {return
_parseRouteResponse(response.body);} else {throw Exception('HTTP
${response.statusCode}');}}
  static String _buildRouteUrl(LatLng from, LatLng to, String
profile) {return
'$_osrmBaseUrl/$profile/'+'${from.longitude.toStringAsFixed(6)},$
{from.latitude.toStringAsFixed(6)};'+'${to.longitude.toStringAsFi
xed(6)},${to.latitude.toStringAsFixed(6)}'?'overview=full&geomet
ries=geojson&steps=true';}static RouteInfo?
_parseRouteResponse(String body) {final data =
json.decode(body);if (data['code'] != 'Ok' ||
data['routes']?.isEmpty ?? true) {throw Exception('Маршрут не
знайдено');}
  final coords = data['routes'][0]['geometry']['coordinates']
as List;final points = coords.map<LatLng?>((c) => (c is List &&
c.length >= 2)? LatLng(c[1].toDouble(), c[0].toDouble()): null)
.whereType<LatLng>() .toList();if (points.isEmpty) throw
Exception
('Порожня геометрія маршруту'); return RouteInfo(points:
points,distance: data['routes'][0]['distance']?.toDouble() ??
0.0,duration: data['routes'][0]['duration']?.toDouble() ??
0.0,);}
  static bool _isValidCoordinate(LatLng point) {
    return point.latitude.abs() <= 90 && point.longitude.abs()
<= 180 && point.latitude != 0.0 && point.longitude != 0.0;}

```

Б.4 Клас LocationService

```

class LocationService {
  static const Duration locationUpdateInterval =
Duration(seconds: 10);
  StreamSubscription<Position>? _positionStream;
  Timer? _locationTimer;
  StreamController<LatLng>? _locationController =
  StreamController<LatLng>.broadcast();
  Stream<LatLng> get locationStream =>
  _locationController?.stream ?? Stream.empty();
  Future<LatLng?> getLocation() async {
  try {if (!await Geolocator.isLocationServiceEnabled())
return null;var permission = await Geolocator.checkPermission();
  if (permission == LocationPermission.denied) {
    permission = await Geolocator.requestPermission();
    if (permission == LocationPermission.denied) return
null;}if (permission == LocationPermission.deniedForever) return
null;final pos = await Geolocator.getCurrentPosition(
  desiredAccuracy: LocationAccuracy.high,
  timeLimit: Duration(seconds: 10),);final location =
LatLng(pos.latitude, pos.longitude);
  _locationController?.add(location);
  return location;} catch (e) {print("Error getting
location: $e");return null;}}

```

Б.5 Клас OneElementScreen

```

class OneElementScreen extends StatelessWidget {
  final ListItem item;
  const OneElementScreen({required this.item});
  @override
  Widget build(BuildContext context) {
    return BlocProvider(
      create: (_) =>
OneElementBloc()..add(FetchOneElement(item.id)),
      child: Scaffold(
        body: Stack(
          children: [
            BlocBuilder<OneElementBloc, OneElementState>(
              builder: (context, state) {
                if (state is OneElementLoading) {
                  return Center(child:
CircularProgressIndicator());} else if (state is
OneElementError) {
                  return Center(child: Text('Error:
${state.message}'));} else if (state is OneElementLoaded) {
                  final item = state.item;
                  return SingleChildScrollView(child:
Column(children: [Image.network(item.imageUrl,width:
double.infinity,height: 400,fit: BoxFit.cover,errorBuilder: (_,
__, ___) => Image.asset('assets/Logo2.png')),
Container(padding:

```

```

EdgeInsets.all(16),decoration: BoxDecoration(color:
Color(0xFFC3E5F5),
                borderRadius:
BorderRadius.vertical(top: Radius.circular(20)),
                boxShadow: [BoxShadow(color:
Colors.black26, blurRadius: 10)], ),child: Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: [Row(children:
[IconButton(icon: Icon(Icons.arrow_back_ios),
                onPressed: () =>
Navigator.pop(context)),),
                Expanded(child: Text(item.title, textAlign:
TextAlign.center, style: TextStyle(fontSize: 20, fontWeight:
FontWeight.bold)),),),],),),SizedBox(height:
15),Text(item.description, textAlign: TextAlign.justify, style:
TextStyle(fontSize: 16)),Divider(),OptionTile(title: 'Перейти до
карти',icon: Icons.pin_drop_outlined,onTap: () =>
Navigator.push(context, MaterialPageRoute(builder: (_) =>
MapScreen(item: item))),),Divider(),OptionTile(title: 'Прокласти
маршрут через Google Maps',icon: Icons.explore_outlined,onTap:
() => openGoogleMaps(item.latitude, item.longitude)),),Divider(),
SizedBox(height: 15), Text("Адреса:", style: TextStyle(fontSize:
16)),SizedBox(height: 10),Text(item.address, style:
TextStyle(fontSize: 16)),if (item.contactInfo?.isNotEmpty ??
false) ...[SizedBox(height: 15), Row(children: [Text('Контакты:
', style: TextStyle(fontSize: 16, fontWeight:
FontWeight.w500)),Flexible(child:
UrlLauncherHelper.buildClickableText(context,item.contactInfo!,d
efaultStyle: TextStyle(fontSize: 16, color:
Colors.black),linkStyle: TextStyle(fontSize: 16, color:
Colors.blue, decoration:
TextDecoration.underline),),),],),),],),),],),),],),),];}return
Container();},),],),),);}}

```