

# AccommodationApp - Project Documentation

Last Updated: January 10, 2025

## Project Overview

AccommodationApp is a comprehensive Android application built with Kotlin and Jetpack Compose that connects students with landlords for accommodation rentals. The app provides a complete ecosystem for property listing, browsing, booking, and management.

## Architecture

- **UI Framework:** Jetpack Compose
- **Language:** Kotlin
- **Database:** SQL Server Express (via JDBC)
- **Architecture Pattern:** MVVM-like with Repositories
- **Navigation:** Jetpack Navigation Compose
- **Async Operations:** Kotlin Coroutines

## User Types & Flows

### 1. Students

#### Features:

- Browse available properties
- Search and filter properties
- View detailed property information
- Save properties to favorites
- Request property bookings with date selection
- View booking history and status
- Manage personal profile
- Contact landlords

#### User Flow:

1. Register → Select University → Complete Profile
2. Browse Properties → Search/Filter → View Details
3. Add to Favorites / Request Booking
4. Track Booking Status (Pending → Approved/Rejected)

## 5. View Profile and Booking History

## 2. Landlords

### Features:

- List and manage properties
- Set availability dates and pricing
- Toggle property active/inactive status
- Review and approve/reject booking requests
- Edit property details
- View booking calendar
- Track property performance
- Manage business profile

### User Flow:

1. Register → Complete Business Profile
2. Add Properties → Set Details and Amenities
3. Manage Bookings → Approve/Reject Requests
4. Edit Properties → Update Availability
5. View Profile and Statistics

## 3. Administrators

### Features:

- Full CRUD operations on all entities
- Manage universities
- Oversee all properties
- Monitor user accounts
- Add/edit amenities
- System-wide oversight

## Key Components

### Data Layer

### Models

- **User** (`user-model.kt`): Base authentication model
  - `userId`, `email`, `passwordHash`, `userType`, `isProfileComplete`

- **Student** (`student-profile-model.kt`): Student-specific profile
  - Personal info, university, budget preferences
- **Landlord** (`landlord-model.kt`): Landlord business profile
  - Business details, verification status, rating
- **Property** (`property-model.kt`): Accommodation listings
  - Details, pricing, availability, amenities, location
- **Booking** (`Booking.kt`): Booking requests and history
  - Dates, status (pending/approved/rejected), pricing
- **University** (`university-model.kt`): University entities
- **Amenity**: Property features and facilities

## Repositories

- **UserRepository**: Authentication and user management
  - Login, registration, profile completion tracking
- **StudentRepository**: Student-specific operations
  - Profile creation and retrieval
- **LandlordRepository**: Landlord-specific operations
  - Profile management, property ownership
- **PropertyRepository**: Property CRUD operations
  - Complex queries, amenity management, status updates
- **BookingRepository**: Booking management
  - Create bookings, check availability, update status
- **FavoritesRepository**: Favorites functionality
  - Add/remove favorites, retrieve favorite properties
- **UniversityRepository**: University data access
- **AdminRepository**: Administrative operations

## Database

- **DatabaseConnection**: JDBC connection management
- **DatabaseConfig**: Connection configuration
- **UserSession**: Global session state management

## UI Layer

### Authentication Screens

- **LoginScreen**: User authentication with role-based routing

- **RegisterScreen:** Account creation with user type selection
- **UniversitySelectionScreen:** University picker for students
- **ProfileCompletionScreen:** Student profile setup
- **LandlordProfileCompletionScreen:** Landlord business profile setup

## Student Screens

- **HomeScreen:** Property browsing with search
  - Search bar functionality
  - Bottom navigation (Home, Bookings, Favorites)
  - Property cards with key information
- **PropertyDetailScreen:** Comprehensive property view
  - Image placeholder, amenities, pricing
  - Favorite toggle functionality
  - Book Now with date selection
  - Landlord information
- **StudentBookingsScreen:** Booking management
  - Tabbed view (All, Pending, Approved, Rejected)
  - Booking cards with status badges
  - Property and date information
- **StudentFavoritesScreen:** Saved properties
  - List of favorited properties
  - Quick remove functionality
  - Navigation to property details
- **StudentProfileScreen:** Personal information
  - Academic details
  - Budget preferences
  - Contact information

## Landlord Screens

- **LandlordHomeScreen:** Property management dashboard
  - Statistics overview (total, active, monthly revenue)
  - Property list with quick actions
  - Active/inactive toggle with warnings
  - Navigation to bookings and profile

- **LandlordAddPropertyScreen:** New property creation
  - Comprehensive property form
  - Amenity selection
  - Date validation
- **LandlordEditPropertyScreen:** Property modification
  - Full edit capabilities
  - Delete with cascade handling
- **LandlordBookingManagementScreen:** Booking requests
  - Tabbed interface (Pending, Approved, History)
  - Approve/Reject functionality
  - Student information display
  - Booking details and messages
- **LandlordProfileScreen:** Business profile
  - Company information
  - Verification status
  - Contact details

## Admin Screens

- **AdminDashboardScreen:** System overview
- **AdminStudentListScreen:** Student management
- **AdminLandlordListScreen:** Landlord oversight
- **AdminPropertyListScreen:** Property administration
- **AdminUniversityListScreen:** University management
- **AdminAddStudentScreen:** Direct student creation

## Shared Components

- **PropertyCard:** Reusable property display component
- **BookingDialog:** Date selection for bookings
- **DatePickerDialog:** Custom date picker
- **StatusBadge:** Booking status visualization

## Navigation

- **Navigation.kt:** Centralized routing configuration
  - Deep linking support
  - Role-based navigation

- Protected routes
- Parameter passing

## Key Features Implementation

### 1. Property Search & Filter

- Real-time search in HomeScreen
- Filters by location, title, and address
- Active properties only for students

### 2. Booking System

- **Date Selection:** Custom date pickers with availability validation
- **Availability Checking:** Prevents double-booking
- **Price Calculation:** Daily rate based on monthly price
- **Approval Workflow:** Landlord review and decision
- **Status Tracking:** Real-time status updates

### 3. Favorites System

- Toggle favorite status from property details
- Persistent storage in database
- Quick access from dedicated screen
- Visual feedback with heart icons

### 4. Property Management

- **CRUD Operations:** Full create, read, update, delete
- **Status Control:** Active/inactive toggling
- **Availability Dates:** From/to date management
- **Amenities:** Many-to-many relationship handling
- **Cascade Deletion:** Proper cleanup of related data

### 5. User Profiles

- **Students:** Academic and preference information
- **Landlords:** Business details and verification
- **Completion Tracking:** Profile completion status

## Database Schema Highlights

- **Normalized Design:** Proper relationships and foreign keys
- **Cascade Operations:** Maintaining referential integrity
- **Audit Fields:** created\_at, updated\_at timestamps
- **Status Enums:** Controlled values for statuses
- **Many-to-Many:** PropertyAmenities junction table

## Security Considerations

- Plain text passwords (TO BE FIXED - implement hashing)
- SQL injection prevention through prepared statements
- Role-based access control
- Session management through UserSession

## Recent Updates (January 10, 2025)

### Student Features Completed:

#### 1. Favorites System

- Add/remove favorites from property details
- Dedicated favorites screen
- Persistent storage with visual feedback

#### 2. Booking System

- Complete booking flow with date selection
- Availability validation
- Price calculation
- Booking history with status tracking

#### 3. Student Profile

- Comprehensive profile display
- Academic and preference information

### Landlord Features Added:

#### 1. Booking Management

- View all booking requests
- Approve/reject functionality
- Organized by status tabs

#### 2. Landlord Profile

- Business information display

- Verification status

## **UI/UX Improvements:**

- Fixed icon compatibility issues
- Improved navigation flow
- Added snackbar feedback
- Enhanced error handling

## **Bug Fixes:**

- Fixed property detail screen duplicate functions
- Resolved date type mismatches
- Fixed repository method naming conflicts
- Corrected navigation parameter passing

## **Future Enhancements**

1. **Image Upload:** Property photos implementation
2. **Messaging System:** Direct landlord-student communication
3. **Reviews & Ratings:** Post-stay feedback system
4. **Payment Integration:** Secure payment processing
5. **Push Notifications:** Booking updates and alerts
6. **Advanced Search:** More filter options (price range, amenities)
7. **Map Integration:** Visual property location
8. **Calendar View:** Availability visualization
9. **Document Upload:** Lease agreements and verification
10. **Multi-language Support:** Internationalization

## **Technical Debt**

1. Implement proper password hashing (bcrypt/scrypt)
2. Add comprehensive error handling
3. Implement proper state management (consider ViewModel)
4. Add unit and integration tests
5. Optimize database queries
6. Implement proper dependency injection
7. Add offline capability
8. Improve date handling consistency



9. Add proper logging framework
10. Implement proper image caching

## Development Notes

- Ensure SQL Server Express is running locally
- Database connection uses Windows Authentication
- All dates stored in ISO format (yyyy-MM-dd)
- Prices stored as DECIMAL(10,2)
- User types: 'student', 'landlord', 'admin'
- Booking statuses: 'pending', 'approved', 'rejected', 'cancelled', 'completed'

## Testing Accounts

- Student: [ana.kovac@student.hr](mailto:ana.kovac@student.hr) / password123
- Landlord: [marko.novak@gmail.com](mailto:marko.novak@gmail.com) / password123
- Admin: (create via direct DB insert)

---

End of Documentation