classDiagram

%% Main Application Controller

class NavigationApp {

-currentUser: User

-activeSession: NavigationSession

-systemSettings: AppSettings

+initialize()

+startNavigation()

+stopNavigation()

+handleEmergency()

}

%% User and Session Management

class User {

-userId: String

-preferences: UserPreferences

-accessibilityNeeds: AccessibilityProfile

+setPreferences(prefs: UserPreferences)

+getAccessibilityProfile(): AccessibilityProfile

}

class NavigationSession {

-sessionId: String

-startTime: DateTime

-currentLocation: Location

-destination: Location

-isActive: boolean

+startSession(destination: Location)

+endSession()

+updateLocation(location: Location)

}

%% Core Navigation Components

class Navigator {

-currentRoute: Route

-mapManager: MapManager

-pathfinder: PathFinder

+calculateRoute(start: Location, end: Location): Route

+updatePosition(location: Location)

+recalculateRoute(): Route

+getNextInstruction(): NavigationInstruction

}

class Route {

-waypoints: List~Location~

-instructions: List~NavigationInstruction~

-totalDistance: float

-estimatedTime: int

+addWaypoint(location: Location)

+getNextWaypoint(): Location

+isComplete(): boolean

}

class NavigationInstruction {

-direction: String

-distance: float

-description: String

-audioContent: String

+generateAudio(): String

+getHapticPattern(): HapticPattern

}

%% Obstacle Detection System

class ObstacleDetector {

-camera: CameraInterface

-visionProcessor: VisionProcessor

-detectionModel: MLModel

-alertThreshold: float

+startDetection()

+stopDetection()

+processFrame(frame: CameraFrame): List~Obstacle~

+calibrateDetection()

}

class Obstacle {

-type: ObstacleType

-position: Position3D

-size: Dimensions

-confidence: float

-timestamp: DateTime

+getDistance(): float

+getDirection(): Direction

+isMoving(): boolean

}

class VisionProcessor {

-openCVEngine: OpenCVEngine

-tensorFlowModel: TensorFlowLite

+preprocessImage(image: CameraFrame): ProcessedImage

+detectObjects(image: ProcessedImage): List~DetectedObject~

+estimateDepth(objects: List~DetectedObject~): List~Obstacle~

}

%% Voice Interface System

class VoiceInterface {

-speechRecognizer: SpeechToText

-speechSynthesizer: TextToSpeech

-nlpProcessor: NLPProcessor

-voiceSettings: VoiceSettings

+processVoiceCommand(audio: AudioInput): Command

+speakInstruction(instruction: String)

+setVoiceSettings(settings: VoiceSettings)

}

class SpeechToText {

-recognitionEngine: STTEngine

-languageModel: LanguageModel

+startListening()

+stopListening()

+convertToText(audio: AudioInput): String

}

class TextToSpeech {

-synthesisEngine: TTSEngine

-voiceProfile: VoiceProfile

+speak(text: String)

+setVoice(voice: VoiceProfile)

+adjustSpeed(speed: float)

+adjustVolume(volume: float)

}

class NLPProcessor {

-intentClassifier: IntentModel

-entityExtractor: EntityModel

+parseCommand(text: String): ParsedCommand

+extractIntent(command: ParsedCommand): Intent

+extractEntities(command: ParsedCommand): List~Entity~

}

%% Landmark Recognition

class LandmarkRecognizer {

-landmarkDatabase: LandmarkDB

-recognitionModel: MLModel

-confidenceThreshold: float

+recognizeLandmark(image: CameraFrame): Landmark

+announceLandmark(landmark: Landmark)

+updateLandmarkDatabase(landmarks: List~Landmark~)

}

class Landmark {

-landmarkId: String

-name: String

-type: LandmarkType

-position: Location

-description: String

-audioDescription: String

+getAudioAnnouncement(): String

+isNearby(userLocation: Location): boolean

}

%% Map and Data Management

class MapManager {

-mapDatabase: SQLiteDB

-currentMap: HospitalMap

+loadMap(mapId: String): HospitalMap

+updateUserLocation(location: Location)

+findNearestLandmarks(location: Location): List~Landmark~

+validatePath(route: Route): boolean

}

class HospitalMap {

-mapId: String

-floorPlans: List~FloorPlan~

-landmarks: List~Landmark~

-walkablePaths: List~Path~

+getFloorPlan(floor: int): FloorPlan

+findPath(start: Location, end: Location): Path

}

class Location {

-x: float

-y: float

-floor: int

-accuracy: float

+distanceTo(other: Location): float

+bearingTo(other: Location): float

}

%% Audio and Feedback Systems

class AudioManager {

-outputDevice: AudioDevice

-inputDevice: MicrophoneDevice

-audioSettings: AudioSettings

+playInstruction(instruction: NavigationInstruction)

+playAlert(alert: AudioAlert)

+recordVoiceCommand(): AudioInput

}

class HapticFeedback {

-hapticEngine: HapticDevice

-patterns: HapticPatternLibrary

+provideTurnFeedback(direction: Direction)

+provideObstacleAlert(obstacle: Obstacle)

+provideDestinationReached()

}

%% Settings and Configuration

class AppSettings {

-audioSettings: AudioSettings

-navigationSettings: NavigationSettings

-accessibilitySettings: AccessibilitySettings

+loadSettings(): AppSettings

+saveSettings()

+resetToDefaults()

}

%% Relationships

NavigationApp "1" -- "1" User

NavigationApp "1" -- "0..1" NavigationSession

NavigationApp "1" -- "1" Navigator

NavigationApp "1" -- "1" ObstacleDetector

NavigationApp "1" -- "1" VoiceInterface

NavigationApp "1" -- "1" LandmarkRecognizer

NavigationApp "1" -- "1" AudioManager

NavigationApp "1" -- "1" AppSettings

Navigator "1" -- "1" MapManager

Navigator "1" -- "0..1" Route

Route "1" -- "\*" NavigationInstruction

Route "1" -- "\*" Location

ObstacleDetector "1" -- "1" VisionProcessor

ObstacleDetector "1" -- "\*" Obstacle

VoiceInterface "1" -- "1" SpeechToText

VoiceInterface "1" -- "1" TextToSpeech

VoiceInterface "1" -- "1" NLPProcessor

LandmarkRecognizer "1" -- "\*" Landmark

MapManager "1" -- "1" HospitalMap

HospitalMap "1" -- "\*" Landmark

HospitalMap "1" -- "\*" Location

AudioManager "1" -- "0..1" HapticFeedback