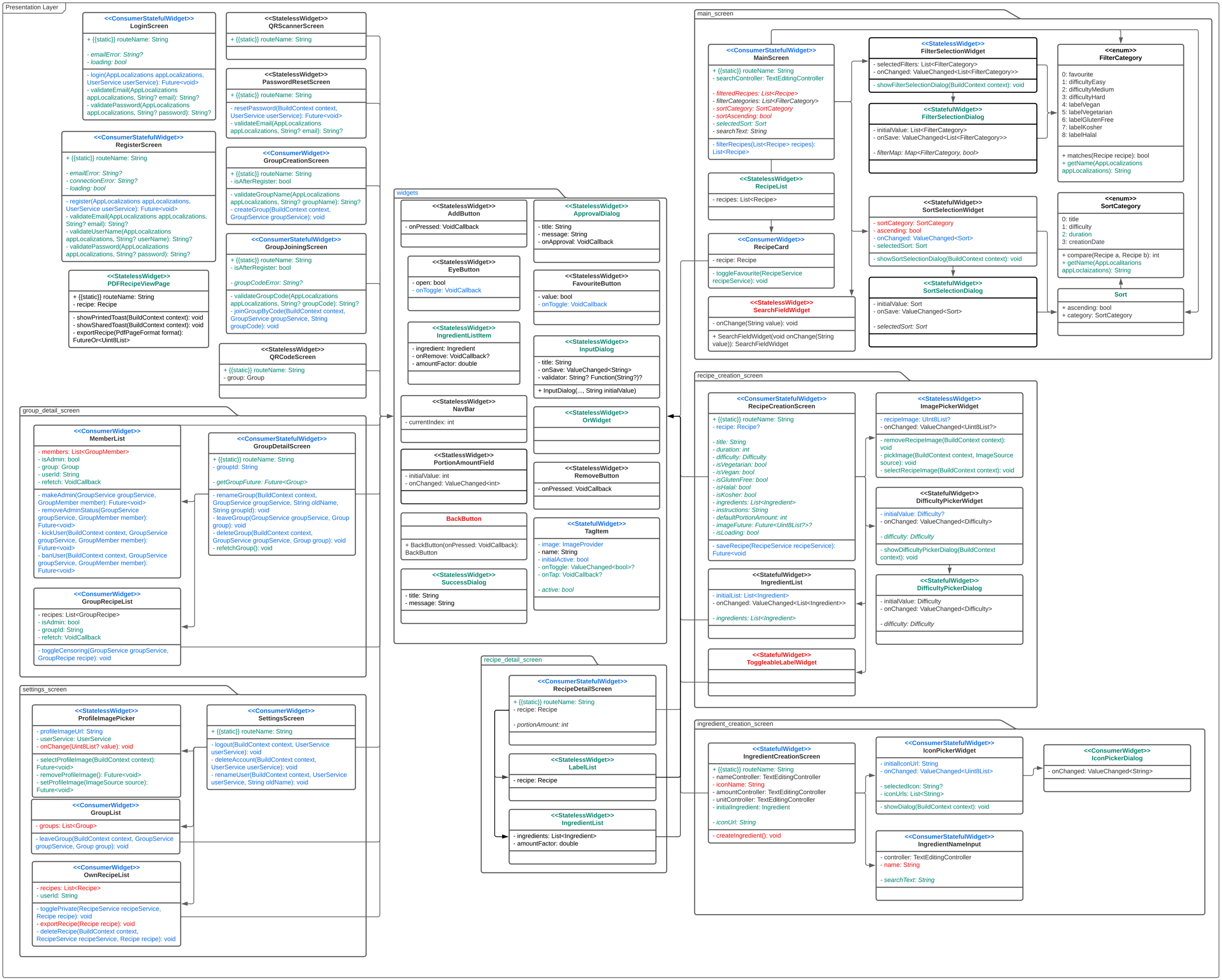
Presentation Layer
<<ConsumerStatefulWidget>> LoginScreen
+ {{static}} routeName: String
- emailError: String?
- loading: bool
- login(AppLocalizations appLocalizations, UserService userService): Future<void>
- validateEmail(AppLocalizations appLocalizations, String? email): String?
- validatePassword(AppLocalizations appLocalizations, String? password): String?
<<StatelessWidget>> QRScannerScreen
+ {{static}} routeName: String
<<StatelessWidget>> PasswordResetScreen
+ {{static}} routeName: String
- resetPassword(BuildContext context, UserService userService): Future<void>
- validateEmail(AppLocalizations appLocalizations, String? email): String?
<<ConsumerStatefulWidget>> RegisterScreen
+ {{static}} routeName: String
- emailError: String?
- connectionError: String?
- loading: bool
- register(AppLocalizations appLocalizations, UserService userService): Future<void>
- validateEmail(AppLocalizations appLocalizations, String? email): String?
- validateUserName(AppLocalizations appLocalizations, String? userName): String?
- validatePassword(AppLocalizations appLocalizations, String? password): String?
<<ConsumerWidget>> GroupCreationScreen
+ {{static}} routeName: String
- isAfterRegister: bool
- validateGroupName(AppLocalizations appLocalizations, String? groupName): String?
- createGroup(BuildContext context, GroupService groupService): void
<<ConsumerStatefulWidget>> GroupJoiningScreen
+ {{static}} routeName: String
- isAfterRegister: bool
- groupCodeError: String?
- validateGroupCode(AppLocalizations appLocalizations, String? groupCode): String?
- joinGroupByCode(BuildContext context, GroupService groupService, String groupCode): void
<<StatelessWidget>> PDFRecipeViewPage
+ {{static}} routeName: String
- recipe: Recipe
- showPrintedToast(BuildContext context): void
- showSharedToast(BuildContext context): void
- exportRecipe(PdfPageFormat format): FutureOr<Uint8List>
<<StatelessWidget>> QRCodeScreen
+ {{static}} routeName: String
- group: Group
main_screen
<<ConsumerStatefulWidget>> MainScreen
+ {{static}} routeName: String
- searchController: TextEditingController
- filteredRecipes: List<Recipe>
- filterCategories: List<FilterCategory>
- sortCategory: SortCategory
- sortAscending: bool
- selectedSort: Sort
- searchText: String
- filterRecipes(List<Recipe> recipes): List<Recipe>
<<StatelessWidget>> RecipeList
- recipes: List<Recipe>
<<ConsumerWidget>> RecipeCard
- recipe: Recipe
- toggleFavourite(RecipeService recipeService): void
<<StatelessWidget>> SearchFieldWidget
- onChange(String value): void
+ SearchFieldWidget(void onChange(String value)): SearchFieldWidget
<<StatelessWidget>> FilterSelectionWidget
- selectedFilters: List<FilterCategory>
- onChanged: ValueChanged<List<FilterCategory>>
- showFilterSelectionDialog(BuildContext context): void
<<StatefulWidget>> FilterSelectionDialog
- initialValue: List<FilterCategory>
- onSave: ValueChanged<List<FilterCategory>>
- filterMap: Map<FilterCategory, bool>
<<enum>> FilterCategory
0: favourite
1: difficultyEasy
2: difficultyMedium
3: difficultyHard
4: labelVegan
5: labelVegetarian
6: labelGlutenFree
7: labelKosher
8: labelHalal
+ matches(Recipe recipe): bool
+ getName(AppLocalizations appLocalizations): String
<<StatefulWidget>> SortSelectionWidget
- sortCategory: SortCategory
- ascending: bool
- onChanged: ValueChanged<Sort>
- selectedSort: Sort
- showSortSelectionDialog(BuildContext context): void
<<enum>> SortCategory
0: title
1: difficulty
2: duration
3: creationDate
+ compare(Recipe a, Recipe b): int
+ getName(AppLocalitarions appLoclaizations): String
<<StatefulWidget>> SortSelectionDialog
- initialValue: Sort
- onSave: ValueChanged<Sort>
- selectedSort: Sort
Sort
+ ascending: bool
+ category: SortCategory
widgets
<<StatelessWidget>> AddButton
- onPressed: VoidCallback
<<StatelessWidget>> ApprovalDialog
- title: String
- message: String
- onApproval: VoidCallback
<<StatelessWidget>> EyeButton
- open: bool
- onToggle: VoidCallback
<<StatelessWidget>> FavouriteButton
- value: bool
- onToggle: VoidCallback
<<StatelessWidget>> IngredientListItem
- ingredient: Ingredient
- onRemove: VoidCallback?
- amountFactor: double
<<StatelessWidget>> InputDialog
- title: String
- onSave: ValueChanged<String>
- validator: String? Function(String?)?
+ InputDialog(..., String initialValue)
<<StatelessWidget>> NavBar
- currentIndex: int
<<StatelessWidget>> OrWidget
<<StatlessWidget>> PortionAmountField
- initialValue: int
- onChanged: ValueChanged<int>
<<StatelessWidget>> RemoveButton
- onPressed: VoidCallback
BackButton
+ BackButton(onPressed: VoidCallback): BackButton
<<StatefulWidget>> TagItem
- image: ImageProvider
- name: String
- initialActive: bool
- onToggle: ValueChanged<bool>?
- onTap: VoidCallback?
- active: bool
<<StatelessWidget>> SuccessDialog
- title: String
- message: String
recipe_creation_screen
<<ConsumerStatefulWidget>> RecipeCreationScreen
+ {{static}} routeName: String
- recipe: Recipe?
- title: String
- duration: int
- difficulty: Difficulty
- isVegetarian: bool
- isVegan: bool
- isGlutenFree: bool
- isHalal: bool
- isKosher: bool
- ingredients: List<Ingredient>
- instructions: String
- defaultPortionAmount: int
- imageFuture: Future<Uint8List?>?
- isLoading: bool
- saveRecipe(RecipeService recipeService): Future<void
<<StatelessWidget>> ImagePickerWidget
- recipeImage: UInt8List?
- onChanged: ValueChanged<Uint8List?>
- removeRecipeImage(BuildContext context): void
- pickImage(BuildContext context, ImageSource source): void
- selectRecipeImage(BuildContext context): void
<<StatefulWidget>> DifficultyPickerWidget
- initialValue: Difficulty?
- onChanged: ValueChanged<Difficulty>
- difficulty: Difficulty
- showDifficultyPickerDialog(BuildContext context): void
<<StatefulWidget>> DifficultyPickerDialog
- initialValue: Difficulty
- onChanged: ValueChanged<Difficulty>
- difficulty: Difficulty
<<StatefulWidget>> IngredientList
- initialList: List<Ingredient>
- onChanged: ValueChanged<List<Ingredient>>
- ingredients: List<Ingredient>
<<StatefulWidget>> ToggleableLabelWidget
group_detail_screen
<<ConsumerWidget>> MemberList
- members: List<GroupMember>
- isAdmin: bool
- group: Group
- userId: String
- refetch: VoidCallback
- makeAdmin(GroupService groupService, GroupMember member): Future<void>
- removeAdminStatus(GroupService groupService, GroupMember member): Future<void>
- kickUser(BuildContext context, GroupService groupService, GroupMember member): Future<void>
- banUser(BuildContext context, GroupService groupService, GroupMember member): Future<void>
<<ConsumerStatefulWidget>> GroupDetailScreen
+ {{static}} routeName: String
- groupId: String
- getGroupFuture: Future<Group>
- renameGroup(BuildContext context, GroupService groupService, String oldName, String groupId): void
- leaveGroup(GroupService groupService, Group group): void
- deleteGroup(BuildContext context, GroupService groupService, Group group): void
- refetchGroup(): void
<<ConsumerWidget>> GroupRecipeList
- recipes: List<GroupRecipe>
- isAdmin: bool
- groupId: String
- refetch: VoidCallback
- toggleCensoring(GroupService groupService, GroupRecipe recipe): void
settings_screen
<<StatelessWidget>> ProfileImagePicker
- profileImageUrl: String
- userService: UserService
- onChange(Uint8List? value): void
- selectProfileImage(BuildContext context): Future<void>
- removeProfileImage(): Future<void>
- setProfileImage(ImageSource source): Future<void>
<<ConsumerWidget>> SettingsScreen
+ {{static}} routeName: String
- logout(BuildContext context, UserService userService): void
- deleteAccount(BuildContext context, UserService userService): void
- renameUser(BuildContext context, UserService userService, String oldName): void
<<ConsumerWidget>> GroupList
- groups: List<Group>
- leaveGroup(BuildContext context, GroupService groupService, Group group): void
<<ConsumerWidget>> OwnRecipeList
- recipes: List<Recipe>
- userId: String
- togglePrivate(RecipeService recipeService, Recipe recipe): void
- exportRecipe(Recipe recipe): void
- deleteRecipe(BuildContext context, RecipeService recipeService, Recipe recipe): void
recipe_detail_screen
<<ConsumerStatefulWidget>> RecipeDetailScreen
+ {{static}} routeName: String
- recipe: Recipe
- portionAmount: int
<<StatelessWidget>> LabelList
- recipe: Recipe
<<StatelessWidget>> IngredientList
- ingredients: List<Ingredient>
- amountFactor: double
ingredient_creation_screen
<<StatefulWidget>> IngredientCreationScreen
+ {{static}} routeName: String
- nameController: TextEditingController
- iconName: String
- amountController: TextEditingController
- unitController: TextEditingController
- initialIngredient: Ingredient
- iconUrl: String
- createIngredient(): void
<<ConsumerStatefulWidget>> IconPickerWidget
- initialIconUrl: String
- onChanged: ValueChanged<Uint8List>
- selectedIcon: String?
- iconUrls: List<String>
- showDialog(BuildContext context): void
<<ConsumerWidget>> IconPickerDialog
- onChanged: ValueChanged<String>
<<ConsumerStatefulWidget>> IngredientNameInput
- controller: TextEditingController
- name: String
- searchText: String