# Trailru Software Design Document

Software Engineering I - CS364 Section 2  
Final Revision

*Contributors:*

|  |  |  |  |
| --- | --- | --- | --- |
| Connor Albright | Dallin Gilbert | Sean Mason | Nick Routsong |
| Adam Applegate | Adam Goff | Jonathan Maxwell | Joseph Sanderson |
| Jared Aston | Robert Hampton | Samantha Neville | Max Schuhmacher |
| Evgeniy Bekker | Ethan Hatch | Dan Ortega | Jacob Smith |
| Reece Brown | Wei Hsu | Jose Paz | Justin Sumida |
| Orion Christensen | Chloe Huang | Spencer Pilkington | Kraig Taylor |
| Trae Folkman | Robert Jones | Riley Preator | Brenton Trebilcock |
| Caleb Georgeson | Leonilson Lopes | Matthew Reed | Joshua Wheeler |

## Executive Summary

*Trailru* is a mobile application for the iOS and Android platforms that targets hikers of all skill levels. It provides detailed information about public hiking spots including community submitted photos, reviews, and customized markers for flagging safety concerns, trail highlights, etc. *Trailru* makes it simple to find a hike that will suit users’ individual needs through advanced filters. By creating an account with *Trailru*, users can save their favorite hikes, track any prior hikes they’ve completed, and submit trail content for the benefit of the community. Our integration with 3rd party equipment suppliers offers a path for revenue in addition to other profit models that can be further explored.

This SDD document has been compiled by students at BYU Idaho for the purpose of defining detailed design and implementation details for the *Trailru* mobile application based on the Trailru Software Requirements Specification (SRS).

## Table of Contents

[**Trailru  
Software Design Document**](#_vamxzx6vuf0b) **1**

[Executive Summary](#_ntqnnqk3mg3a) 2

[Table of Contents](#_7sdcyq6iq5sy) 3

[List of Figures](#_756xmqpi226v) 13

[Definitions](#_w6uzlon2owju) 17

[References](#_fqer21rtoaf3) 17

[Traceability Matrix](#_m5y2l2t55zb4) 18

[**5.2 Context**](#_uov0bk5h9eau) **19**

[5.2.1 Keyword search](#_br3qkhwkmplo) 19

[5.2.2 Hike recommendations](#_zfvrkoigub99) 20

[5.2.3 Check Hiker traffic](#_ffto4nejq7xd) 21

[5.2.4 Add hikes to app](#_c9li9gvcrv39) 21

[5.2.5 Hike Distance and Difficulty](#_kg7jfak6pf5w) 22

[5.2.6 Hike classification](#_s1a59lq7pesb) 22

[5.2.7 Browse photos of hike](#_okgyp4cm5y8h) 23

[5.2.8 Download map coordinates](#_vftg0ygbmnhp) 23

[5.2.9 Flag inappropriate content](#_sbtmoswfj26r) 24

[5.2.10 View images in a lightbox](#_s22caajvko1p) 25

[5.2.11 View often used equipment](#_30m7pj7j5l8t) 26

[5.2.12 Indicate Appropriateness](#_8en4s0diq3q9) 26

[5.2.13 Mark Trail Map](#_ast9c0tx0nv) 27

[5.2.14 Access Phone Camera](#_d3tjtob2y7ug) 28

[5.2.15 Enable Emergency Assistance Access](#_vt3t7juvho0k) 28

[5.2.16 Count Steps and Track Health](#_oy365qnna1z3) 29

[5.2.17 System interfaces](#_cl1f87fmlmyg) 30

[5.2.18 User interfaces](#_l2ttjqf41hfc) 31

[5.2.19 Trail map](#_ovj95vriyksc) 32

[5.2.20 Trail information](#_xmppt8yfmmrg) 33

[5.2.21 Trail Review System](#_o644i753ik32) 34

[5.2.22 Trail safety](#_16y6ehtycio0) 35

[5.2.23 Trail search](#_djaitxjlz6dc) 36

[5.2.24 Hike recommendations](#_edw8o0523mr6) 37

[5.2.25 Hiking gear](#_wwlc5zo3juch) 38

[5.2.26 Trail map](#_fpqo9mb5v57i) 38

[5.2.27 Trail information](#_etrqqoqtrxsp) 39

[5.2.28 Trail review system](#_za3arixk41ae) 40

[5.2.29 Trail safety](#_kkzto0bl2sh6) 41

[**5.3 Composition**](#_9kcq59g375ks) **42**

[5.3.1 High-Level Application Package Composition (SRS 1.3.1, 3.6 & SDD 5.7)](#_3zn7x4sz5xw2) 42

[5.3.2 Application-Side Component Elaboration (SRS 3.6.5, 1.3.1, 1.3.3, 3.1.1, 3.6)](#_tukbg5ni5lwq) 43

[5.3.2.1 React Native (SDD 5.7.1)](#_cva8cvvxpuw7) 43

[5.3.2.1.1 React Native Benefit](#_bo49ljjh771j) 43

[5.3.2.2 User Interface (SDD 5.8, 5.10.1)](#_582cmye1n8s) 43

[5.3.2.3 Phone Interface (SDD 5.8, 5.6.1, 5.6.4)](#_p1otg9hawr6g) 44

[5.3.3 Server-Side Component Elaboration (SRS 1.3.1, 3.1.2.1.4 & SDD 5.6.2)](#_i89shxah70ho) 44

[5.3.3.1 Apache](#_hudc6qil3p09) 44

[5.3.3.2 MySQL (SDD 5.6.2.1)](#_yfk8p88dyb2a) 44

[5.3.4 External Interfaces Elaboration (SRS 1.3.1, 3.1, 3.6.4 & SDD 5.5.2, 5.6.3)](#_i8lvu4qo26o3) 45

[5.3.4.1 Google Maps API (SRS 1.3.1.4, 3.1.2.1.3)](#_vu70h8nueq8y) 45

[5.3.4.2 National Parks API(SRS 3.1.2.1.1)](#_4p3ccikew801) 45

[5.3.4.3 Advertisements (SRS 3.1.2.1.2)](#_wpiwa698jidg) 45

[5.3.4.4 OpenWeatherMap API (SRS 3.6.4)](#_250nm3iayfla) 45

[5.3.4.5 Emergency Services API (SRS 3.1.2.1.6)](#_nvwu19clcrom) 46

[5.3.4.6 Local businesses (SRS 3.1.2.1.5)](#_z3n4syf2253) 46

[**5.4 Logical**](#_xn3s8tujn3sd) **46**

[5.4.1 Trail Class (SRS 1.2.2-1.2.8, 3.2.2)](#_glzkiuvmk6j2) 47

[5.4.1.1 Trail Name](#_2btkhvo4hu4w) 47

[5.4.1.2 Trail ID](#_7r4y0hl55td1) 47

[5.4.1.3 Difficulty](#_na4gk2myiaem) 48

[5.4.1.4 Distance](#_n757q5wdc1pv) 48

[5.4.1.5 Age Range](#_vt367db0ufzs) 48

[5.4.1.6 Pets](#_xyy864h40vcu) 48

[5.4.1.7 Trail Type](#_cyrs5s5ibgat) 48

[5.4.1.8 Description](#_uzhv0u81hkfc) 48

[5.4.1.9 Get Trail Name](#_h4vz8fvpyf4j) 48

[5.4.1.9 Set Trail Name](#_eqgm9mja1uf3) 48

[5.4.1.10 Get Trail ID](#_14q6ycybiytg) 48

[5.4.1.10 Set Trail ID](#_4afj1dqd30yh) 49

[5.4.1.11 Get Difficulty](#_xdaekvmkr21o) 49

[5.4.1.11 Set Difficulty](#_tj01r8njkjhg) 49

[5.4.1.12 Get Distance](#_qhbskk5salnq) 49

[5.4.1.12 Set Distance](#_wewcttj82ed4) 49

[5.4.1.13 Get Age Range](#_t3owfyxacm6d) 49

[5.4.1.13 Set Age Range](#_70wfcqd7laj0) 49

[5.4.1.14 Pets Allowed](#_13hb2jdvcfmv) 49

[5.4.1.14 Set Pets Allowed](#_z7n9wb1cyfc2) 49

[5.4.1.15 Get Trail Type](#_js4kv6kys3ah) 50

[5.4.1.15 Set Trail Type](#_jb5u3kuigyte) 50

[5.4.1.16 Get Description](#_1xulmv3u7xq2) 50

[5.4.1.16 Set Description](#_bi7ipuxoc931) 50

[5.4.1.17 Add Trail Photo](#_mpwq08hpv5z3) 50

[5.4.1.17 Get Trail Photo](#_w7uwb0aajnik) 50

[5.4.2 User Class (SRS 3.2.6)](#_c0z5h2kk7l6f) 51

[5.4.2.1 User Id](#_702f6qrut9q4) 51

[5.4.2.2 Username](#_2lmnnjcru7jp) 51

[5.4.2.3 Token](#_q0bcxhgu8jj) 51

[5.4.2.4 Email](#_kysv7ewtyeb) 51

[5.4.2.5 Edit User](#_a9ttln539vey) 52

[5.4.2.6 Delete User](#_n913khub9s71) 52

[5.4.2.7 Get Username](#_x9kv8gpb765b) 52

[5.4.2.8 Set Username](#_deh96hdnkk2) 52

[5.4.2.10 Get Token](#_s7p0wk88dz9o) 52

[5.4.2.11 Set Token](#_pdzc7ygpwfv3) 52

[5.4.3 Feedback Class (SRS 3.4.3, SRS 3.2.3)](#_9hurh09h2t68) 52

[5.4.3.1 Get Id](#_kqc7xxf4b8s5) 53

[5.4.3.2 Set Id](#_sr4xkx4k5az5) 53

[5.4.3.3 Get Trail Id](#_6sxs4iewq6k4) 53

[5.4.3.4 Set Trail Id](#_tyoam6qjdvql) 53

[5.4.3.5 Set User Id](#_v3wcsqa8om65) 53

[5.4.3.6 Get User Id](#_2tr28f9jj175) 53

[5.4.3.7 Get Body](#_1by2cdxuvixy) 54

[5.4.3.8 Set Body](#_m4erpse6sufx) 54

[5.4.3.1 Review Class (SRS 3.4.3)](#_yummshe41dim) 54

[5.4.3.1.1 Add Review](#_v4ugm4y9ek1z) 54

[5.4.3.1.2 Delete Review](#_yn0s1qfj7cxr) 54

[5.4.3.1.3 Edit Review](#_ikgp71ke8xt) 54

[5.4.3.1.4 Get Title](#_i4b50j3moihn) 55

[5.4.3.1.5 Set Title](#_eljbk3gz7cvb) 55

[5.4.3.1.6 Get Rating](#_gha1fibm8g7w) 55

[5.4.3.1.7 Set Rating](#_xss7b3qyoxjj) 55

[5.4.3.2 Comment Class (SRS 3.2.3)](#_gt1gjcocorm4) 55

[5.4.3.2.1 Add Comment](#_h5um8syddpqc) 55

[5.4.3.2.2 Edit Comment](#_ciqlkytillex) 56

[5.4.3.2.3 Delete Comment](#_4h5rfbkp7iw8) 56

[5.4.3.2.4 Get Parent Id](#_8xpm1hfdc0gz) 56

[5.4.3.2.5 Set Parent Id](#_4u20yjo8m4yb) 56

[5.4.5 Map Class (SRS 1.2.8, 3.2.1, 3.6.4)](#_kw5o2wr7yc2j) 56

[5.4.5.1 Longitude](#_tncpqfalx1pz) 57

[5.4.5.2 Latitude](#_hjne9i65aeo4) 57

[5.4.5.3 Pin Points](#_3lavudh4ijo6) 57

[5.4.5.4 Terrain Description](#_yqlhbi9yjhxv) 57

[5.4.5.5 Terrain Difficulty](#_8uzjn81u2vxi) 57

[5.4.5.6 Get Longitude](#_qmp1ulxyxor3) 57

[5.4.5.7 Set Longitude](#_m4j0bbci3t4) 57

[5.4.5.8 Get Latitude](#_5nq2bgs21u3x) 57

[5.4.5.9 Set Latitude](#_395uv7dq7fmn) 57

[5.4.5.10 Get Pinpoints](#_53s1yma8lrpp) 57

[5.4.5.11 Add Pinpoint](#_qvohizcgn5e5) 57

[5.4.5.12 Delete Pinpoint](#_mpnl2clf291j) 57

[5.4.5.13 Get Terrain Description](#_4sqgt6ox17ta) 57

[5.4.5.14 Set Terrain Description](#_fvv3znygi8wk) 58

[5.4.5.15 Get Terrain Difficulty](#_e9g8g8ftqeo0) 58

[5.4.5.16 Set Terrain Difficulty](#_qxshb4wryf3o) 58

[5.4.6 Pinpoint Class (SRS 1.2.8, 3.2.1)](#_31mn9rripp74) 58

[5.4.6.1 Longitude](#_4hy83kuo74sq) 58

[5.4.6.2 Latitude](#_jh9jicredyv9) 59

[5.4.6.3 Color](#_9d7h21e36xzu) 59

[5.4.6.4 Id](#_jt1fx8bjgh8h) 59

[5.4.6.5 Get Longitude](#_c9l92mq4y3aq) 59

[5.4.6.6 Set Longitude](#_epyuijp0fevn) 59

[5.4.6.7 Get Latitude](#_k7fxve761e93) 59

[5.4.6.8 Set Latitude](#_x8k0mwvjyf68) 59

[5.4.6.9 Get Color](#_ncm7zg8520d1) 59

[5.4.6.10 Set Color](#_qlt2zccnha8x) 59

[5.4.6.11 Get Id](#_1p8qwefdpczs) 59

[5.4.6.12 Set Id](#_hqcnxlizqe37) 59

[5.4.7 Trail List Class](#_rgad1dapq6rk) 59

[5.4.7.1 Trail List](#_jtwyykib23q2) 60

[5.4.7.2 Trail](#_flgu8yuv36x5) 60

[5.4.7.3 Trail List Id](#_qgwmz3otgnln) 60

[5.4.7.4 Get Trail List](#_z2zj1simvfos) 60

[5.4.7.5 Set Trail List](#_xy6zf59h1qky) 60

[5.4.7.6 Get Trail](#_t0vubinc5srd) 61

[5.4.7.7 Set Trail](#_8zdtnqjjzszk) 61

[5.4.7.8 Remove Trail](#_n9rayc7ujfnb) 61

[5.4.7.9 Add Trail](#_f3c3akuwp6r2) 61

[5.4.7.10 Get Trail List Id](#_f3lytdh6wf8t) 61

[5.4.7.11 Set Trail List Id](#_a3ugw0qn01lj) 61

[**5.5 Dependency**](#_ir58hixas5av) **61**

[5.5.1 UML Component Entities](#_9474rvovhj5q) 61

[5.5.2 UML Component Diagrams](#_vxgr0gglc1lh) 62

[5.5.2.1 Camping Information API](#_otx5s74xyzlj) 62

[5.5.2.2 Advertising API](#_mo1i1jrevon5) 62

[5.5.2.3 Maps API](#_4mhvm9jjvo6z) 63

[5.5.2.4 Trailru REST API](#_d95o45n4gk6w) 63

[5.5.2.5 Local Businesses API](#_704vcgm8gmlh) 64

[5.5.2.6 Safety API](#_h1c6artxmcyn) 65

[5.5.2.7 Weather API](#_6dws3alt3e2l) 66

[**5.6 Information**](#_n1kafurt71u5) **66**

[5.6.1 Local Application/Client Database](#_ydtphvoy9xs0) 67

[5.6.1.1 Local Database over SQLite](#_n8drguds6pdz) 67

[5.6.1.2 Local Database Information Encryption](#_bqm26tbyif5f) 67

[5.6.1.3 Local Data Layer Diagram](#_3mxb082o58lk) 68

[5.6.2 Back End / Server Side Database](#_km2f2zbg9e2b) 69

[5.6.2.1 Server Side Database over MySQL 8.0+](#_i6e8vcplwo8g) 69

[5.6.2.2 Database Entities definition](#_hpibnlumc43j) 70

[5.6.2.3 Database Entities Relationship Diagram](#_llai4rxzmbpg) 73

[5.6.3 GPS](#_ax5u5bpzrbt) 75

[5.6.3.1 GPS functionality](#_czth0fhx8rta) 75

[5.6.3.3 My Location Figure](#_gxae0aana4a8) 77

[5.6.4 Map and Data Caching](#_smfkmqmzc350) 77

[5.6.4.1 Cached Data](#_nb1oyqx1s2vr) 77

[**5.7 Patterns**](#_kyglrfoyg8pa) **78**

[5.7.1 React Native Cross-Platform Communication](#_y91pnozhr55j) 78

[5.7.2 Model View Controller Design Pattern](#_vzhknxv2qgk1) 79

[5.7.2.1 Diagram](#_62ma4fpsan4b) 80

[**5.8 Interface**](#_dmferlgss3qu) **80**

[5.8.1 Home Page](#_wrk4yiu4siiy) 82

[5.8.1.1 Keyword Search (SRS 1.2.1)  
 A display that has a list of popular keywords that describe hikes.](#_3tig2wb5vtbj) 82

[5.8.1.1.2 Hike Recommendations (SRS 3.2.7)](#_vic6ufa92e3a) 82

[5.8.1.3 Favorites Button (SRS 3.2.6.1)](#_yiqe3pxfxna7) 83

[5.8.1.3.1 Save as Favorite](#_jzotb1hds6fu) 83

[5.8.1.3.2 Favorites Button](#_90hxyn8fzb1y) 84

[5.8.1.4 Hike Card (SRS 3.2.2)](#_2xfcsvqnlhgd) 84

[5.8.2 Hike Page](#_zhwzkip4pat4) 85

[5.8.2.1 Hike Photos](#_ltrfegx3v0t3) 85

[5.8.2.2 Hike Difficulty](#_j45gi7dqzj1i) 86

[5.8.2.3 Hiker Traffic](#_jb15zqkerzo5) 87

[5.8.2.4 Hike Classification](#_rkoeikpfursl) 88

[5.8.2.5 Download Map (SRS 3.4.1)](#_lhnnjpdw5cxu) 89

[5.8.2.6 Flag Inappropriate Content (SRS 3.4.3)](#_pkgv9tfzto8c) 90

[5.8.2.7 Weather Information (SRS 3.6.4)](#_w1ynqtokfiux) 91

[5.8.3 Camera Page](#_7d3rutdm8ta6) 92

[5.8.3.1 Camera View (SRS 1.2.14)](#_zf56gofwfcxx) 92

[5.8.3.2 Switch Camera Button](#_wj00aolt7bb) 92

[5.8.3.3 Take Picture Button](#_7e6wzak8423o) 92

[5.8.4 Write Review Page](#_fgcqbtqo4yvo) 93

[5.8.4.1 Review panel (SRS 3.2.3)](#_pndmvgwie419) 93

[5.8.4.2 Submit Review Button (SRS 3.2.3)](#_aps1zt92lz2g) 94

[**5.9 Structure**](#_m3pxewmv08u3) **94**

[5.9.1 Trail Class (SRS 1.2.2-1.2.8, 3.2.2)](#_v5l1viaxo54k) 94

[5.9.2 User Class (SRS 3.2.6)](#_dc98ucyvxjh1) 95

[5.9.3 Review Class (SRS 1.2.12, 3.4.3)](#_dr6jdm3mzzog) 96

[5.9.4 Comment Class (SRS 3.2.3)](#_7euyk7jqk5kr) 97

[5.9.5 Map Class (SRS 1.2.8, 3.2.1, 3.6.4)](#_v5l1viaxo54k) 97

[5.9.6 Pinpoint Class (SRS 1.2.8, 3.2.1)](#_v5l1viaxo54k) 98

[5.9.7 Trail List Class](#_3g4bb7pm6x9u) 98

[5.9.8 Feedback Class](#_1jk5ylvlvu2d) 99

[**5.10 Interaction**](#_j5x2l971jbhh) **100**

[5.10.1 User Interaction](#_3vc1rndpv5w2) 100

[5.10.1.1 Trail Search (SRS 3.2.5, SDD 5.8.1.1)](#_xstuonsodu0i) 100

[5.10.1.2 Trail Recommendations (SRS 3.2.7, SDD 5.8.1.1.2)](#_21z70t6snh18) 101

[5.10.1.3 Add Hikes (SRS 3.2.5.4)](#_dx55jmr8xz5i) 102

[5.10.1.4 Check Hiker Traffic (SRS 3.2.2)](#_c5qp5isvlq70) 103

[5.10.1.5 Hike Distance and Difficulty (SRS 3.2.3)](#_jzletj34p7m5) 104

[5.10.1.6 Hike Information (SRS 3.2.2)](#_26t6q7hoa5fn) 104

[5.10.1.7 Download Map Coordinates (SRS 3.2.1.1)](#_1bgdptnvi7dm) 104

[5.10.2 Trail Safety (SRS 3.2.4)](#_wqmh4ui381nj) 105

[5.10.2.1 Weather Information (SRS 3.2.4.1)](#_k1jgv7vdrdy6) 105

[5.10.2.2 Danger Information (SRS 3.2.4.3)](#_hx2i7bpioqdd) 106

[5.10.2.3 Rules and Regulations (SRS 3.2.4.4, SRS 3.2.4.6)](#_41ei5hm6ir97) 108

[5.10.2.4 Emergency Messages (SRS 3.2.4.7)](#_w63mvjoykjrj) 109

[5.10.2.5 Emergency Contacts (SRS 3.2.4.2, SRS 3.2.4.5)](#_afnebhvemuge) 110

[5.10.3 Device Permissions](#_mlf9tplv74r7) 112

[5.10.3.1 Requesting Camera Permissions](#_1keskdnlmylf) 112

[**5.11 State Dynamics**](#_apaqd3eefsda) **113**

[5.11.1 Design Entities](#_ag62f3b8y1od) 113

[5.11.1.1 Outline of Entities and Their Activities](#_w3yah5ib2j6) 113

[5.11.2 State Dynamics Views](#_brnjorsyx59i) 115

[5.11.2.1 Home Page](#_gnnxm2njf1oh) 115

[5.11.2.2 Recommendations Page](#_qsz80v6qby9y) 116

[5.11.2.3 Favorites List](#_siuk4goccru) 116

[5.11.2.4 Search Results](#_j7q469dwkxg8) 117

[5.11.2.5 Location Page](#_btundshiuyw) 118

[**5.12 Algorithm**](#_ntufdc78fhm7) **119**

[5.12.1 Hike Recommendation Algorithm (SRS 1.2.2, 3.2.7, 3.4.6)](#_dtwi9sunycrn) 120

[5.12.1.1 Rater Class](#_k7itc9fzw3jr) 121

[5.12.1.2 Similar Class](#_e4aubqxobrle) 122

[5.12.1.2.1 byUser Function](#_386y63w1k6es) 123

[5.12.1.3 Suggestion Class](#_cntfyo13v6j9) 124

[5.12.1.3.1 forUser Function](#_5xfdvnpa3kcw) 124

[5.12.2 Keyword Search](#_285dxwjv4c21) 125

[5.12.3 Trail Search](#_rorkzhucapxh) 125

[**5.13 Resources**](#_qzuncwd388u9) **126**

[5.13.1 Search results in 5 seconds or less](#_swn4xh8tfc9b) 126

[5.13.2 Traffic level view in 3 seconds or less](#_d1kjv6jasxta) 127

[5.13.3 Hike distance and difficulty view in 3 seconds or less](#_i3576hqjynvr) 127

[5.13.4 Maps shall be downloaded in 30 seconds or less](#_50m007exrj06) 128

[5.13.5 Hiking equipment recommendations in 5 seconds or less](#_n06u3pm5peip) 128

[5.13.6 Phone camera access in 3 seconds or less](#_xfb7do63dda1) 129

[5.13.7 Connection to API’s is done in 3 second or less](#_sscpx94pv0e4) 129

## 

## 

## List of Figures

|  |  |
| --- | --- |
| **Figure** | **Description** |
| [5.2.1.1](#ixaideytlk8f) | Keyword Search Diagram |
| [5.2.2.1](#2qg4a6h4pj5a) | Hike Recommendations Diagram |
| [5.2.3.1](#7f971fyosq32) | Check Hiker Diagram |
| [5.2.4.1](#sns1dzvwaz74) | Adding Hikes to App Diagram |
| [5.2.5.1](#5d2d394a5gg) | Hike Distance and Difficulty Diagram |
| [5.2.6.1](#qrwdq0tybvey) | Hike Classification Diagram |
| [5.2.7.1](#dgidnnp0tm5) | Browsing Photos Diagram |
| [5.2.8.1](#6lpodoxar5j) | Download Map Coordinates Diagram |
| [5.2.9.1](#fyvu2x9oikju) | Flag Inappropriate Content Diagram |
| [5.2.10.1](#9l679v6dgfdu) | View Images in a Lightbox Diagram |
| [5.2.11.1](#kbm45v7wzfhx) | View Often-Used Equipment Diagram |
| [5.2.12.1](#4p3bb7za64v5) | Indicate Appropriateness Diagram |
| [5.2.13.1](#6a13jcrdxdnw) | Mark Trail Map Diagram |
| [5.2.14.1](#jcsrxlmehtq7) | Access Phone Camera Diagram |
| [5.2.15.1](#c81npvkkklgu) | Enable Emergency Assistance Access Diagram |
| [5.2.16.1](#mlcpgbe0cgth) | Count Steps and Track Healthier Diagram |
| [5.2.17.1](#8roxm25im2s2) | System Interfaces Diagram |
| [5.2.18.1](#xkjptbafoddo) | User Interfaces Diagram |
| [5.2.19.1](#rofzx123bt9q) | Trail Map Diagram |
| [5.2.20.1](#juahi9yd4zn) | Trail Information Diagram |
| [5.2.21.1](#nex8j9dwewfg) | Trail Review System Diagram |
| [5.2.22.1](#vqzjkbgkhrzo) | Trail Safety Diagram |
| [5.2.23.1](#8f1lubeh7foq) | Trail Search Diagram |
| [5.2.24.1](#75j8ww4u3a5b) | Hike Recommendations Diagram |
| [5.2.25.1](#oqwd7d5jwt) | Hiking Gear Diagram |
| [5.2.26.1](#1p0j6ziftxl3) | Trail Map Diagram |
| [5.2.27.1](#i10firjq4o3o) | Trail Information Diagram |
| [5.2.28.1](#aneazx7uam4g) | Trail Review System Diagram |
| [5.2.29.1](#qma1gffa8et4) | Trail Safety Diagram |
| [5.3.1](#_3zn7x4sz5xw2) | High Level Application Package Diagram |
| [5.3.2](#_tukbg5ni5lwq) | Application Side Package Diagram |
| [5.3.3](#_i89shxah70ho) | Server Side Package Diagram |
| [5.3.4](#_i8lvu4qo26o3) | External Interfaces Package Diagram |
| [5.4.1](#_glzkiuvmk6j2) | Trail Class UML |
| [5.4.2](#7ogx51fdh933) | User Class UML |
| [5.4.3](#e03cvft9j52s) | Feedback Class UML |
| [5.4.3.1](#1dvcwntpk25r) | Review Class UML |
| [5.4.3.2](#sk32nlaedrau) | Comment Class UML |
| [5.4.4](#75ftb92iws5q) | Map Class |
| [5.4.5](#pmxvokrhazft) | Pinpoint Class |
| [5.4.6](#n5131qmkc9of) | Trail List Class |
| [5.5.2.1.1](#jc6wzm9jr437) | Component diagram for Camping API |
| [5.5.2.1.2](#xh7q9xbpksuk) | Component diagram for Advertising API |
| [5.5.2.1.3](#xch5mhioz7fz) | Component diagram for Maps API |
| [5.5.2.1.4](#cisba9f94tby) | Component diagram for Trailru REST API |
| [5.5.2.1.5](#g5ngcus4mrua) | Component diagram for Local Businesses API |
| [5.5.2.1.6](#9jtxsu40ktw8) | Component diagram for Safety API |
| [5.5.2.1.7](#ve45eefdky5u) | Component diagram for Weather API |
| [5.8.1](#r6drglf7nhib) | User Interface Mock-Up |
| [5.8.1.1.1](#6yzx68rf3lic) | Search Box |
| [5.8.1.2.1](#cp6zinohg1zm) | Hike Recommendations |
| [5.8.1.3.1.1](#857ublgnqrks) | Save as Favorite Button |
| [5.8.1.3.2.1](#jz7dlnpecigh) | Favorites Button |
| [5.8.1.4.1](#is6weazjnfw) | Hike Card |
| [5.8.2.1.1](#3zfk6r81f6o6) | Hike Photos |
| [5.8.2.2.1](#f047rhn1mn8t) | Hike Difficulty |
| [5.8.2.3.1](#2dkdw1t70sxc) | Hike Traffic |
| [5.8.2.4.1](#xhdkas6afbit) | Hike Classification |
| [5.8.2.5.1](#6zybsmjn6cl7) | Download Map |
| [5.8.2.6.1](#9hpbytrbx14j) | Flag Inappropriate Content |
| [5.8.2.7.1](#qf9lywgryj7q) | Weather Information |
| [5.8.3.1](#msldw55kfkwm) | Camera View |
| [5.8.4.1.1](#8sv7ct61u9ux) | Review Panel |
| [5.9.1](#23d2kdqma67o) | Trail Entity Relationship Diagram |
| [5.9.2](#8010bucvu3pr) | User Entity Relationship Diagram |
| [5.9.3](#yzaiup3u75x0) | Review Entity Relationship Diagram |
| [5.9.4](#37pp4olq5l2d) | Comment Entity Relationship Diagram |
| [5.9.5](#agzrgey4cmmv) | Map Entity Relationship Diagram |
| [5.9.6](#z3acu1i76ovr) | Pinpoint Entity Relationship Diagram |
| [5.9.7](#9kqpn2i7bg03) | Trail List Entity Relationship Diagram |
| [5.9.8](#av316f2uy2vw) | Feedback Entity Relationship Diagram |
| [5.10.1.1.1](#ygip9nwdaaiu) | Trail Search |
| [5.10.1.2.1](#fmdjg23jabl) | Save User Data |
| [5.10.1.2.2](#zh6f79bbh2pp) | Request Hike Recommendations |
| [5.10.1.3.1](#nwodgpffowbw) | Add Hikes |
| [5.10.1.4.1](#cp9gkee8rd77) | Check Hiker Traffic |
| [5.10.1.6.1](#dbvgvwwc4ayw) | Request Hike Information |
| [5.10.1.7.1](#mcqhlh3fdep8) | Download Map Coordinates |
| [5.10.2.1.1](#tw48uxkmcjyl) | Request Weather Information |
| [5.10.2.2.1](#808pk9snz92x) | Register a Danger |
| [5.10.2.2.2](#m8i8c6tl012n) | Request Dangers |
| [5.10.2.3.1](#8wg62ou31z2h) | Register Rules |
| [5.10.2.3.2](#ilpvot2pylid) | Request Rules |
| [5.10.2.4.1](#715ywtskhv49) | Send Emergency Call |
| [5.10.2.5.1](#hey5lv921xhn) | Register Emergency Number |
| [5.10.2.5.2](#deow12eiv05o) | Send Emergency Message |
| [5.10.3.1.1](#ez65luvu8ydg) | Request Camera Permissions |
| [5.11.2.1.1](#kix.b6qphphp89j8) | Home Page Diagram |
| [5.11.2.2.1](#kix.1pazcqt49mt8) | Recommendations Page Diagram |
| [5.11.2.3.1](#kix.tnnep1yoytuz) | Favorites List Diagram |
| [5.11.2.4.1](#kix.k8zm37sfljaq) | Search Results Diagram |
| [5.11.2.5.1](#kix.meso0j8526zp) | Location Page Diagram |
| [5.12.1](#cugbmysn8c7i) | Flowchart depicting the logical flow of the Trail Recommendation Algorithm |
| [5.12.1.1](#h5ms5mvdma31) | Diagram of the Rater class |
| [5.12.1.2](#7qemwg9hl7t) | Diagram of the Similar class |
| [5.12.1.3](#xp8sskbhhl24) | Diagram of the Suggestion class |
| [5.13.1.1](#kix.oljc5uxht5h) | Search Results Diagram |
| [5.13.2.1](#kix.1lcgvpqpudxf) | Traffic Level View Diagram |
| [5.13.3.1](#kix.mshqk0p1papl) | Hike Distance and Difficulty Diagram |
| [5.13.4.1](#kix.1z4h4p2wcs07) | Map Downloads Diagram |
| [5.13.5.1](#kix.1ltgwoumfwp8) | Hiking Equipment Recommendations Diagram |
| [5.13.6.1](#kix.eiirg8ojtqff) | Phone Camera Access Diagram |
| [5.13.7.1](#kix.qdyaf3isg02c) | API Connection Diagram |

## Definitions

|  |  |
| --- | --- |
| **Word or Phrase** | **Definition** |
| Component Entity | Specific component within the component design |
| Component Design | How all component entities relate to and utilize one another |
| Component Diagram | Visual representation of the component design |
| Similarity Index | Float ranging from -1.0 to 1.0. The closer the value is to 1.0, the greater the similarity between the two users and their interests. |
|  |  |
|  |  |
|  |  |
|  |  |

## References

|  |  |  |
| --- | --- | --- |
| **Reference Title** | **Publisher** | **Source** |
| Predicting Likes: Inside a Simple Recommendation Engine’s Algorithms | Toptal | https://www.toptal.com/algorithms/predicting-likes-inside-a-simple-recommendation-engine |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Traceability Matrix

[External reference to Traceability Matrix](https://docs.google.com/spreadsheets/u/0/d/112DFmImicsY5vv96p0D7g8eao7DJaKuoIQU3n_Mu_-M/edit)

# 5.2 Context

According to page 14, section 5.2 of the IEEE 1016-2009 standard, “The Context viewpoint depicts services provided by a design subject with reference to an explicit context. That context is defined by reference to actors that include users and other stakeholders, which interact with the design subject in its environment.”

The context viewpoint design elements in the Trailru application include use cases, user and information exchange flow (descriptions, charts, etc.), triggers, primary scenarios, alternate scenarios, pre-conditions, and post-conditions.

This section contains diagrams and entities that describe different active elements interacting with the design such as users, other stakeholders and external systems (actors) and its associated services (internal behavior outlined in [Section 5.11](#_36vfiblmgq4q)) that captures flow of information of the context exchanged.

### 5.2.1 Keyword search

|  |
| --- |
|  |
| *Figure 5.2.1.1: Keyword Search Diagram* |

|  |  |
| --- | --- |
| Summary | A text box where the user can input keywords and the system returns back a list of matches. |
| Actor | User |
| Primary Scenario | The user interacts with the keyword search icon on the home screen |
| SRS Requirement | [SRS 1.2.1](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.8.1.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.u8tczi), [SDD 5.12.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.14hx32g) |

##### 

### 5.2.2 Hike recommendations

|  |
| --- |
|  |
| *Figure 5.2.2.1 Hike Recommendations Diagram* |

|  |  |
| --- | --- |
| Summary | Ability of the app to list hikes that might be enjoyable based on the users previous hikes and liked hikes |
| Actor | User |
| Primary Scenario | User views hikes and suggestions are automatically shown to user |
| Pre-Conditions | There is a base of past hikes liked by the user |
| Assumptions | The user has liked hikes in the database |
| SRS Requirement | [SRS 1.2.2](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79), [SRS 3.2.7](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79), [SRS 3.4.6](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.8.1.1.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.u8tczi), [SDD 5.10.1.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5), [SDD 5.11.2.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3b2epr8), [SDD 5.12.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.14hx32g) |

### 5.2.3 Check Hiker traffic

|  |
| --- |
|  |
| *Figure 5.2.3.1 Check Hiker Traffic Diagram* |

|  |  |
| --- | --- |
| Summary | A screen that lists how popular and well traveled the hike is. |
| Actor | User |
| Primary Scenario | The user navigates to a trail information page |
| SRS Requirement | [SRS 1.2.3](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.8.2.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp), [SDD 5.10.1.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5), [SDD 5.13.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.is565v) |

##### 

### 5.2.4 Add hikes to app

|  |
| --- |
|  |
| *Figure 5.2.4.1 Adding Hikes to App Diagram* |

|  |  |
| --- | --- |
| Summary | Functionality to add trails to the app. |
| Actor | User |
| Primary Scenario | The user attempts to add a hike to *Trailru*’s database |
| SRS Requirement | [SRS 1.2.4](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.4.7.9](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3c9z6hx), [SDD 5.10.1.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5) |

##### 

### 5.2.5 Hike Distance and Difficulty

|  |
| --- |
|  |
| *Figure 5.2.5.1 Hike Distance and Difficulty Diagram* |

|  |  |
| --- | --- |
| Summary | A screen that lists the difficulty details and attributes of the selected hike. |
| Actor | User |
| Primary Scenario | The user navigates to a trail information page and attempts to view its content |
| SRS Requirement | [SRS 1.2.5](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.4.1.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3s49zyc), [SDD 5.4.1.12](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.1d96cc0), [SDD 5.8.2.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp), [SDD 5.10.1.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5), [SDD 5.13.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.32rsoto) |

##### 

### 5.2.6 Hike classification

|  |
| --- |
|  |
| *Figure 5.2.6.1 Hike Classification Diagram* |

|  |  |
| --- | --- |
| Summary | A screen that lists the details about the hike. |
| Actor | User |
| Primary Scenario | The user navigates to a trail information page and attempts to view its content |
| SRS Requirement | [SRS 1.2.6](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.8.2.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp) |

### 5.2.7 Browse photos of hike

|  |
| --- |
|  |
| *Figure 5.2.7.1 Browsing Photos Diagram* |

|  |  |
| --- | --- |
| Summary | A screen where people who have hiked the trail can post photos they took while on the trail. |
| Actor | User |
| Primary Scenario | The user navigates to a trail information page and attempts to view its content |
| SRS Requirement | [SRS 1.2.7](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.4.1.17](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.243i4a2), [SDD 5.8.2.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp), [SDD 5.11.1.1.9](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.452snld) |

##### 

### 5.2.8 Download map coordinates

|  |
| --- |
|  |
| *Figure 5.2.8.1 Download Map Coordinates Diagram* |

|  |  |
| --- | --- |
| Summary | Functionality to get coordinates in an offline situation. |
| Actor | User |
| Primary Scenario | The user attempts to download a map for offline viewing |
| SRS Requirement | [SRS 1.2.8](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.3.4.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2250f4o), [SDD 5.4.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.4hr1b5p), [SDD 5.5.1.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.15phjt5), [SDD 5.5.2.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.xevivl), [SDD 5.6.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3mj2wkv), [SDD 5.8.2.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp), [SDD 5.9.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3e8gvnb), [SDD 5.10.1.7](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5), [SDD 5.13.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.1hx2z1h), [SDD 5.13.7](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.w7b24w) |

##### 

### 5.2.9 Flag inappropriate content

### 

|  |
| --- |
|  |
| *Figure 5.2.9.1 Flag Inappropriate Content Diagram* |

|  |  |
| --- | --- |
| Summary | Functionality to notify content as unhelpful, listed in the wrong place or just inappropriate |
| Actor | User |
| Primary Scenario | The user attempts to flag content as inappropriate |
| SRS Requirement | [SRS 1.2.9](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.8.2.6](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp) |

##### 

### 5.2.10 View images in a lightbox

|  |
| --- |
|  |
| *Figure 5.2.10.1 View Images in a Lightbox Diagram* |

|  |  |
| --- | --- |
| Summary | Images can be shown within a lightbox to remove distractions. |
| Actor | User |
| Primary Scenario | The user selects an image to be viewed within a lightbox |
| SRS Requirement | [SRS 1.2.10](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.11.1.1.9](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.452snld) |

##### 

### 5.2.11 View often used equipment

Ability to check what equipment other hikers used on a hike

|  |
| --- |
|  |
| *Figure 5.2.11.1 View Often-Used Equipment Diagram* |

|  |  |
| --- | --- |
| Summary | Ability to check what equipment other hikers used on a hike |
| Actor | User |
| Primary Scenario | The user navigates to a trail information page and attempts to view its content |
| SRS Requirement | [SRS 1.2.11](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.10.1.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5), [SDD 5.13.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.41wqhpa) |

##### 

### 5.2.12 Indicate Appropriateness

|  |
| --- |
|  |
| *Figure 5.2.12.1 Indicate Appropriateness Diagram* |

|  |  |
| --- | --- |
| Summary | Functionality to see if a hike is appropriate for children or pets |
| Actor | User |
| Primary Scenario | The user navigates to a trail information page and indicates whether the trail is appropriate for children or pets |
| SRS Requirement | [SRS 1.2.12](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.4.1.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.279ka65), [SDD 5.4.1.6](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.meukdy), [SDD 5.4.1.14](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.rjefff) |

##### 

### 5.2.13 Mark Trail Map

|  |
| --- |
|  |
| *Figure 5.2.13.1 Mark Trail Map Diagram* |

|  |  |
| --- | --- |
| Summary | Functionality to add map markers and notes to a map |
| Actor | User |
| Primary Scenario | The user uses a pin to mark a point on a trail’s map |
| SRS Requirement | [SRS 1.2.13](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.6.2.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.393x0lu) |

##### 

### 5.2.14 Access Phone Camera

### 

|  |
| --- |
|  |
| *Figure 5.2.14.1 Access Phone Camera Diagram* |

|  |  |
| --- | --- |
| Summary | Functionality to access the smartphone camera within the *Trailru* app instead of backing out to use the camera from the GUI desktop |
| Actor | User |
| Primary Scenario | The user attempts to take a picture of a trail that can later be attached to the information page of the respective trail |
| SRS Requirement | [SRS 1.2.14](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.8.3.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp), [SDD 5.10.3.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3rnmrmc), [SDD 5.13.6](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2h20rx3) |

##### 

### 5.2.15 Enable Emergency Assistance Access

|  |
| --- |
|  |
| *Figure 5.2.15.1 Enable Emergency Assistance Access Diagram* |

|  |  |
| --- | --- |
| Summary | When a hiker becomes in trouble or has an accident, emergency assistance is enabled and the ability to notify assistance is accessible from the app. |
| Actor | User |
| Primary Scenario | A situation arises where the user is in need of assistance. The user then attempts to contact emergency services |
| SRS Requirement | [SRS 1.2.15](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.3.4.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.40ew0vw), [SDD 5.5.2.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2i9l8ns), [SDD 5.10.2.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3rnmrmc) |

##### 

### 5.2.16 Count Steps and Track Health

|  |
| --- |
|  |
| *Figure 5.2.16.1 Count Steps and Track Healthier Diagram* |

|  |  |
| --- | --- |
| Summary | Functionality added to see how many steps are added and other health tracking implements |
| Actor | User |
| Primary Scenario | The user walks on a trail. Their steps are counted. |
| Alternate Scenario | The user attempts to view the app’s current step count |
| SRS Requirement | [SRS 1.2.16](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.3.2.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.48pi1tg) |

##### 

### 5.2.17 System interfaces

|  |
| --- |
|  |
| *Figure 5.2.17.1 System interfaces Diagram* |

|  |  |
| --- | --- |
| Summary | The app has the ability to be run on multiple platforms. |
| Actor | User |
| Primary Scenario | The user goes to install *Trailru* onto his/her device. |
| Assumptions | The user’s device runs on Android or iOS. |
| SRS Requirement | [SRS 1.3.1.1](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.7.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.30tazoa), [SDD 5.13.7](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.w7b24w) |

##### 

### 5.2.18 User interfaces

|  |
| --- |
|  |
| *Figure 5.2.18.1 User interfaces Diagram* |

|  |  |
| --- | --- |
| Summary | The user’s phone’s operating system determines the layout of screens, widgets, and navigation controls. |
| Actor | User |
| Primary Scenario | The user launches the app and views the screen. |
| SRS Requirement | [SRS 1.3.1.2](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.3.2.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.1opuj5n), [SDD 5.7.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.30tazoa), [SDD 5.7.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.1fyl9w3), [SDD 5.8](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp), [SDD 5.10.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5) |

##### 

### 5.2.19 Trail map

### 

|  |
| --- |
|  |
| *Figure 5.2.19.1 Trail Map Diagram* |

|  |  |
| --- | --- |
| Summary | Ability to call up a map of the trail for navigation while hiking. |
| Actor | User, Google Maps API |
| Primary Scenario | The user accesses a map, either from Google’s database or from local storage. |
| Alternate Scenario | The user shall be able to download the map that is not currently in local storage. |
| Exception Scenario | The user has no network connection nor has the map in local storage. |
| SRS Requirement | [SRS 3.2.1](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.3.4.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2250f4o), [SDD 5.4.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.4hr1b5p), [SDD 5.5.1.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.15phjt5), [SDD 5.5.2.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.xevivl), [SDD 5.6.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3mj2wkv), [SDD 5.8.2.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp), [SDD 5.9.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3e8gvnb), [SDD 5.10.1.7](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5), [SDD 5.13.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.1hx2z1h), [SDD 5.13.7](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.w7b24w) |

##### 

### 5.2.20 Trail information

|  |
| --- |
|  |
| *Figure 5.2.20.1 Trail Information Diagram* |

|  |  |
| --- | --- |
| Summary | Ability to call up information about a trail |
| Actor | User |
| Primary Scenario | Information about a trail is to be displayed to the user |
| Pre-Conditions | This information exists and is reachable by the user |
| SRS Requirement | [SRS 3.2.2](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.5.2.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.33zd5kd), [SDD 5.10.1.6](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5) |

### 5.2.21 Trail Review System

|  |
| --- |
|  |
| *Figure 5.2.21.1 Trail Review System Diagram* |

|  |  |
| --- | --- |
| Summary | Functionality to post what a hiker thought about a trail. |
| Actor | User |
| Primary Scenario | The user attempts to post a review of a trail |
| Alternate Scenario | The user attempts to flag the trail or review for inappropriate content |
| Assumptions | The user has been to and experienced the trail |
| SRS Requirement | [SRS 3.2.3](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.4.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.odc9jc), [SDD 5.8.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp), [SDD 5.9.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.4ddeoix) |

##### 

### 5.2.22 Trail safety

|  |
| --- |
|  |
| *Figure 5.2.22.1 Trail Safety Diagram* |

|  |  |
| --- | --- |
| Summary | Functionality to view safety rules and guidelines unique to a particular trail as well as general hiking/camping safety/rules. |
| Actor | User, Other Hikers, Emergency Contact, Search and Rescue |
| Primary Scenario | The user attempts to view trail safety information |
| Alternate Scenario | The user attempts to report emergency information or wild animal activity |
| Pre-Conditions | The user has location services enabled |
| SRS Requirement | [SRS 3.2.4](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.5.2.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2i9l8ns), [SDD 5.10.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3rnmrmc) |

##### 

### 5.2.23 Trail Search

|  |
| --- |
|  |
| *Figure 5.2.23.1 Trail Search Diagram* |

|  |  |
| --- | --- |
| Summary | Ability of the app to search for trails |
| Actor | User |
| Primary Scenario | The user types in a place or hike to find a hike around them |
| Pre-Conditions | The user is connected to the internet and the app database |
| SRS Requirement | [SRS 3.2.5](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79), [SRS 3.4.5](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.8.1.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.u8tczi), [SDD 5.10.1.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5), [SDD 5.11.1.1.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.452snld), [SDD 5.11.2.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.4a7cimu), [SDD 5.12.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.14hx32g), [SDD 5.12.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.14hx32g), [SDD 5.13.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.23muvy2) |

##### 

### 5.2.24 Hike recommendations

|  |
| --- |
|  |
| *Figure 5.2.24.1 Hike Recommendations Diagram* |

|  |  |
| --- | --- |
| Summary | Ability of the app to list hikes that might be enjoyable based on the users previous hikes and liked hikes |
| Actor | User |
| Primary Scenario | User views hikes and suggestions are automatically shown to user |
| Pre-Conditions | There is a base of past hikes liked by the user |
| Assumptions | The user has liked hikes in the database |
| SRS Requirement | [SRS 3.2.7](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79), [SRS 3.4.6](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.8.1.1.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.u8tczi), [SDD 5.10.1.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5), [SDD 5.11.2.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3b2epr8), [SDD 5.12.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.14hx32g) |

##### 

### 5.2.25 Hiking gear

|  |
| --- |
|  |
| *Figure 5.2.25.1 Hiking Gear Diagram* |

|  |  |
| --- | --- |
| Summary | Ability to access a list of hiking gear based on the trail or hike the user is currently evaluating |
| Actor | User |
| Primary Scenario | User wants to know what kind of gear would be useful for the hike |
| Pre-Conditions | User is currently looking at a hike |
| Assumptions | The user has no previous knowledge of gear needed. |
| SRS Requirement | [SRS 3.2.8](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79), [SRS 3.4.7](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.10.1.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5), [SDD 5.13.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.41wqhpa) |

##### 

### 5.2.26 Trail map

|  |
| --- |
|  |
| *Figure 5.2.26.1 Trail Map Diagram* |

|  |  |
| --- | --- |
| Summary | Ability to access the trail map and view details about the trail |
| Actor | User |
| Primary Scenario | User wants to view the full map of the hike |
| SRS Requirement | [SRS 3.4.1](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.3.4.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2250f4o), [SDD 5.4.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.4hr1b5p), [SDD 5.5.1.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.15phjt5), [SDD 5.5.2.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.xevivl), [SDD 5.6.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3mj2wkv), [SDD 5.8.2.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp), [SDD 5.9.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3e8gvnb), [SDD 5.10.1.7](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5), [SDD 5.13.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.1hx2z1h), [SDD 5.13.7](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.w7b24w) |

### 5.2.27 Trail information

|  |
| --- |
|  |
| *Figure 5.2.27.1 Trail Information Diagram* |

### 

|  |  |
| --- | --- |
| Summary | Ability to view key features and description of the trail |
| Actor | User |
| Primary Scenario | The user wants to view the key features and description |
| Pre-Conditions | User is viewing a hike |
| SRS Requirement | [SRS 3.4.2](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.5.2.1](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.33zd5kd), [SDD 5.10.1.6](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.26sx1u5) |

##### 

### 5.2.28 Trail review system

|  |
| --- |
|  |
| *Figure 5.2.28.1 Trail Review System Diagram* |

|  |  |
| --- | --- |
| Summary | Ability to save the hike to the user’s account |
| Actor | User |
| Primary Scenario | User wants to save their current hike to their account |
| SRS Requirement | [SRS 3.4.3](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.4.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.odc9jc), [SDD 5.8.4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2f3j2rp), [SDD 5.9.3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.4ddeoix) |

##### 

### 5.2.29 Trail safety

|  |
| --- |
|  |
| *Figure 5.2.29.1 Trail Safety Diagram* |

|  |  |
| --- | --- |
| Summary | Ability of the app to show a list of safety concerns about the hike |
| Actor | User |
| Primary Scenario | The user viewing a hike will be able to see safety concerns |
| Assumptions | There is safety information to be shown to the user |
| SRS Requirement | [SRS 3.4.4](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |
| SDD References | [SDD 5.5.2.5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.2i9l8ns), [SDD 5.10.2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#heading=h.3rnmrmc) |

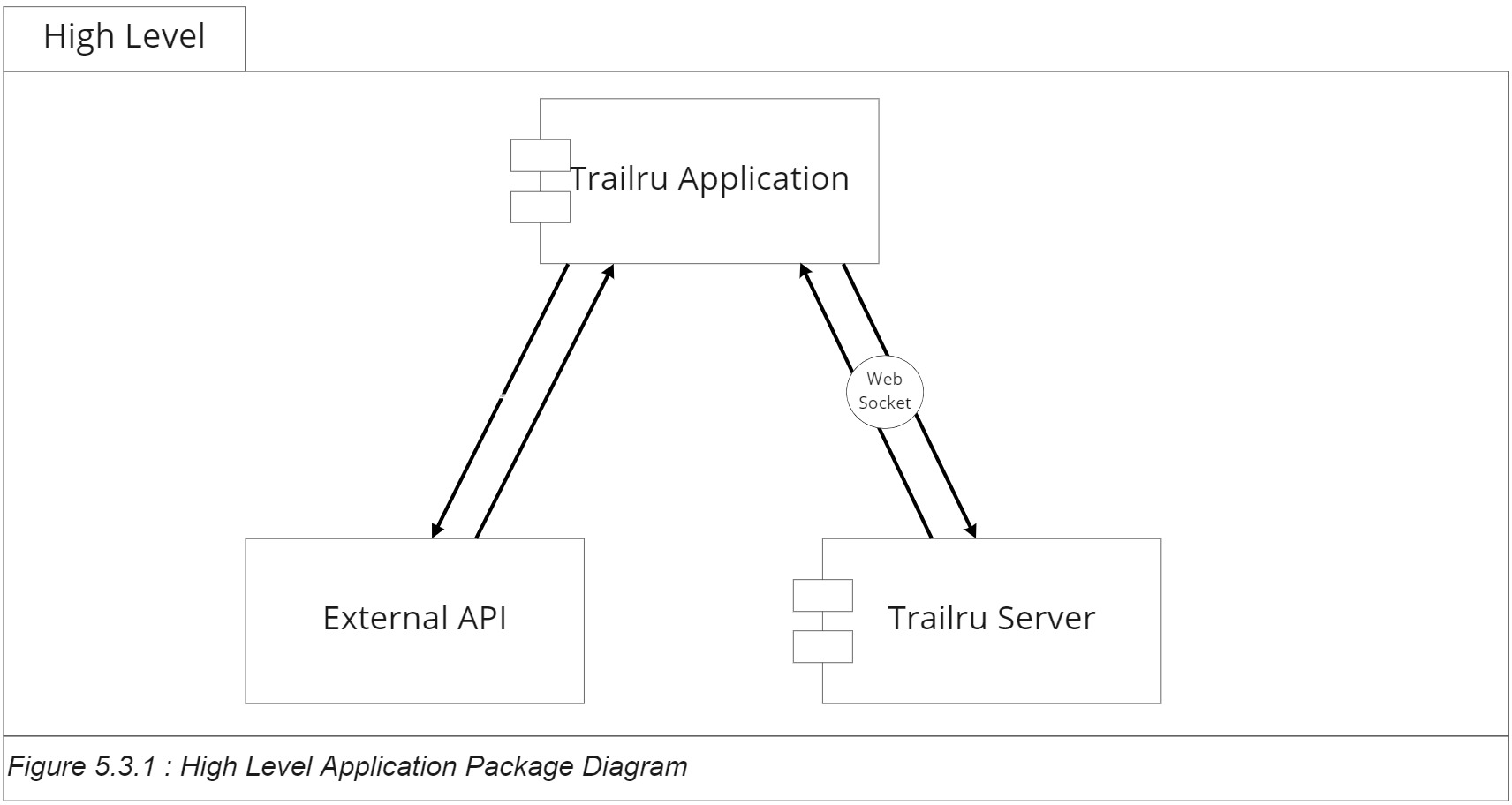
# 

# 5.3 Composition

The composition viewpoint is described in IEEE Std 1016-2009 on pages 15-16, section 5.3: “The Composition viewpoint describes the way the design subject is (recursively) structured into constituent parts and establishes the roles of those parts.”

### 5.3.1 High-Level Application Package Composition (SRS [1.3.1](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [3.6](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit) & SDD [5.7](#_kyglrfoyg8pa))

The following diagrams depict the high level view for the system composition of the *Trailru* application, the Trailru Server (API), and the External APIs.

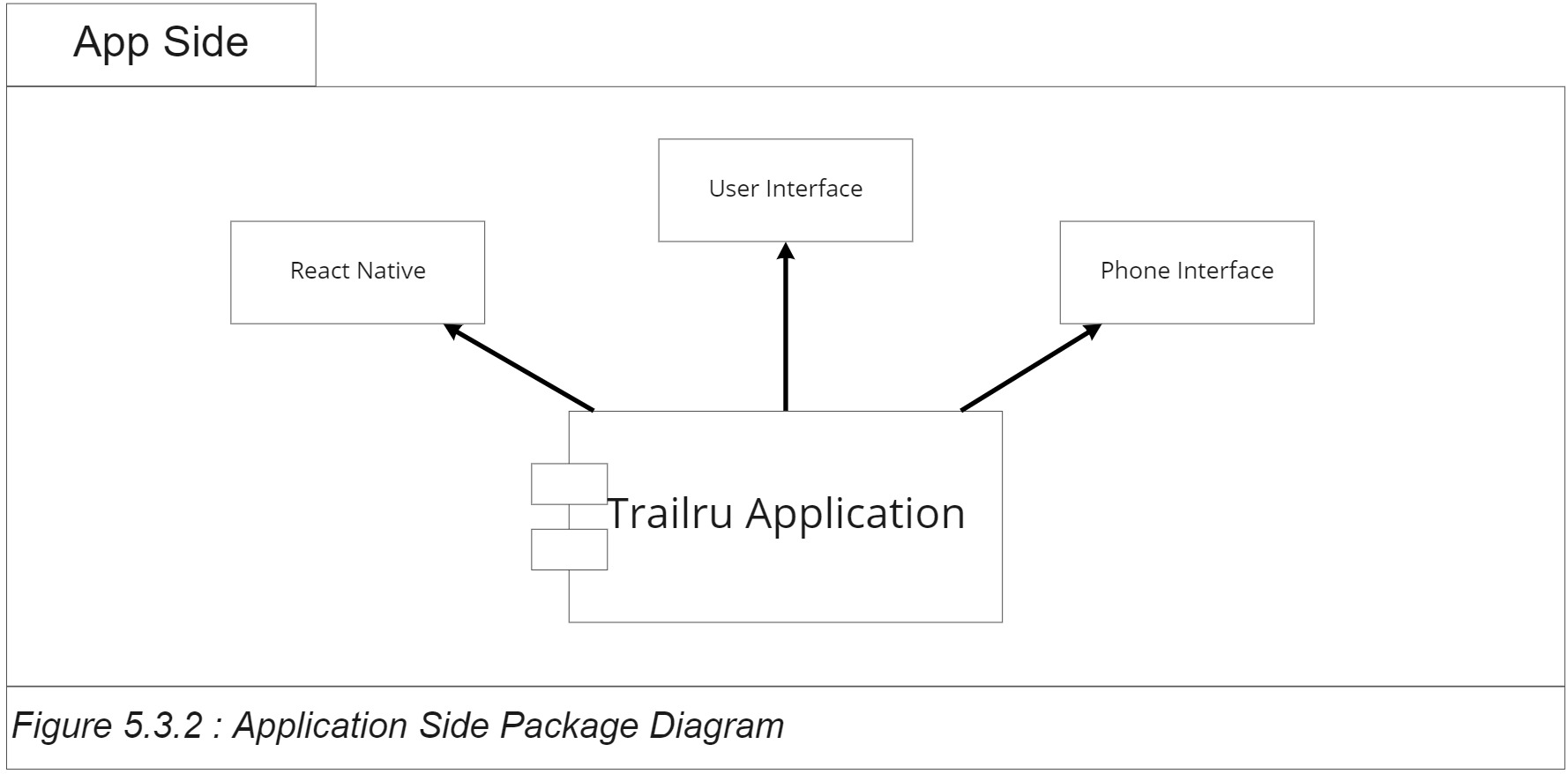
[](#pg743w8xd58g)

The Trailru app consists of two pieces of software: the application, and the Trailru Server (API). The server is built on a Linux OS running Apache, with MySQL to manage and communicate with the application and the database.

The Trailru mobile application will be built with React Native, using the Android and the iOS development platforms.

### 5.3.2 Application-Side Component Elaboration (SRS [3.6.5](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [1.3.1](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [1.3.3](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [3.1.1](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [3.6](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

The application-side components consist of the Software Development Kit, the user interface, and the phone interface. These three sections make up the role of the application within the context of the high-level application package composition.

[](#s7owysdxjuoy)

#### 5.3.2.1 React Native (SDD [5.7.1](#_y91pnozhr55j))

The application will handle interactions with the user's device and calls to various servers through the React JavaScript framework. React has certain advantages:

##### 5.3.2.1.1 React Native Benefit

React is a write once, run anywhere development platform. The virtual DOM in react-based programs is similar to HTML documents. The user experience is easier to design for and app development is easier to track and manage. The data-flow model of React creates stable applications, which leads to less crashing.

#### 5.3.2.2 User Interface (SDD [5.8](#_dmferlgss3qu), [5.10.1](#_3vc1rndpv5w2))

The application will maintain connection with the user interface, meaning any type of multi-tasking function the phone operating system has, as well as the touch screen interface and home button (if applicable).

The application will respond appropriately while in airplane mode (no access to external data) and do not disturb mode (no sounds or alerts).

#### 5.3.2.3 Phone Interface (SDD [5.8](#_dmferlgss3qu), [5.6.1](#_ydtphvoy9xs0), [5.6.4](#_smfkmqmzc350))

The application will interact with the phone storage and system events. This includes the phone battery level, as well as any health telemetry the phone receives. The battery level will be used to establish guidelines between the low-power mode and the full-power mode.

### 5.3.3 Server-Side Component Elaboration (SRS [1.3.1](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [3.1.2.1.4](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit) & SDD [5.6.2](#_km2f2zbg9e2b))

The Trailru server will handle user login and authentication that will be sent back to the application through the web socket. The server will utilize a Linux based OS, Apache for communications, and MySQL for storage of user names and hashed passwords. The server will receive a request and transmit back data after Apache and the server fulfill the request.

### 

#### 5.3.3.1 Apache

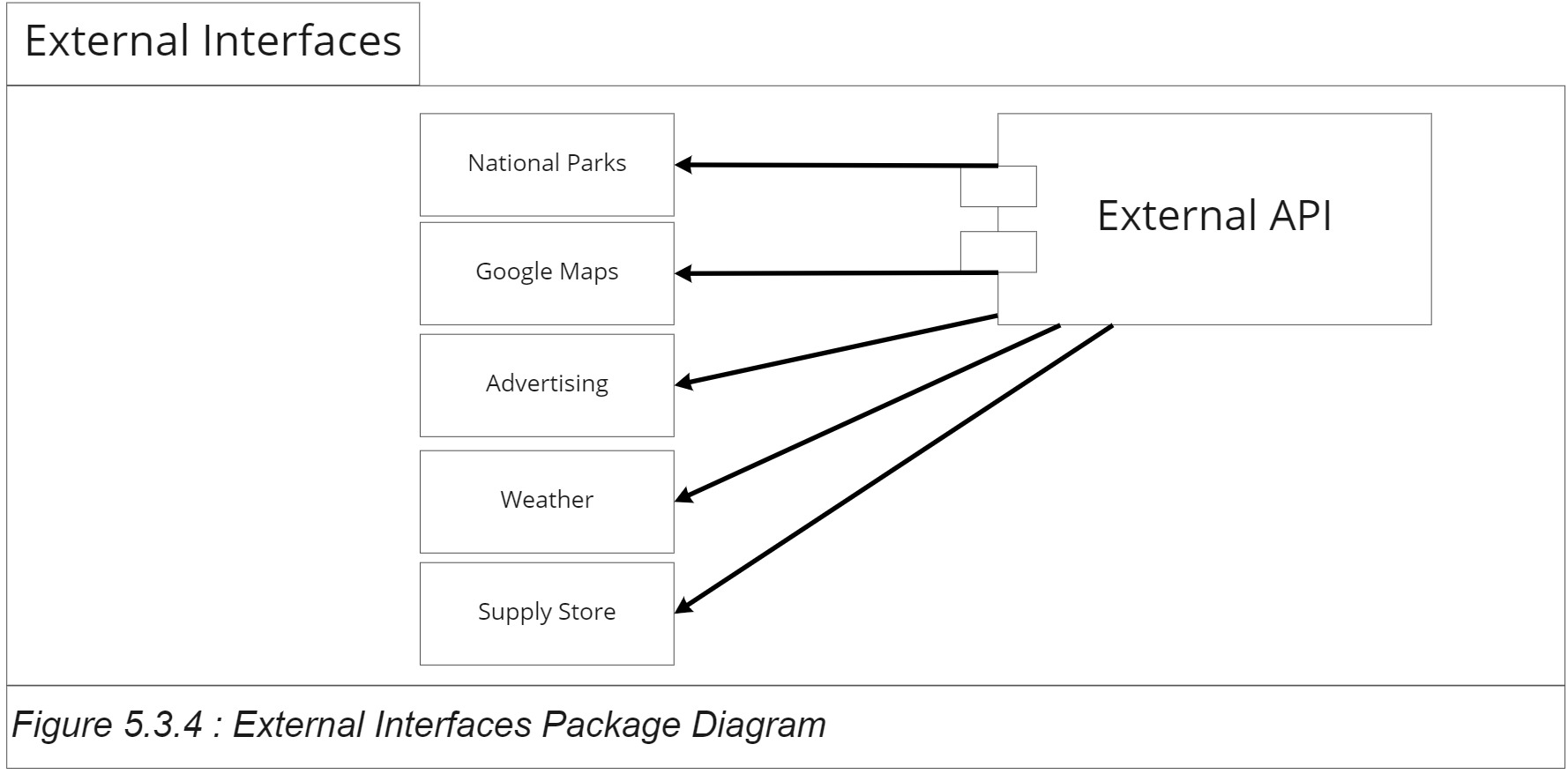
The Apache HTTP server is an open source HTTP server for UNIX and Windows based operating systems. The Trailru server will send requests to Apache httpd 2.4.46. The Apache server will then respond with the appropriate files.

#### 5.3.3.2 MySQL (SDD [5.6.2.1](#_i6e8vcplwo8g))

Refer to SDD 5.6.2.1 for more information regarding MySql.

### 5.3.4 External Interfaces Elaboration (SRS [1.3.1](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [3.1](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [3.6.4](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit) & SDD [5.5.2](#_vxgr0gglc1lh), [5.6.3](#_ax5u5bpzrbt))

Trailru will interact with external interfaces through the usage of APIs and services. It will send and receive data to APIs and request data from local services and advertisements. The following figure elaborates further the external interfaces used in Trailru.

[](#edgmvng8wvle)

#### 5.3.4.1 Google Maps API (SRS [1.3.1.4,](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit) [3.1.2.1.3](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit) & SDD [5.5.2.3](#_4mhvm9jjvo6z))

Google Maps will be used to implement the map functions and location based services for the Trailru application.

#### 5.3.4.2 National Parks API(SRS [3.1.2.1.1](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit) & SDD [5.5.2.1](#_otx5s74xyzlj))

National Parks API will provide the application data for current parks, trails, campsites, and reservations.

#### 5.3.4.3 Advertisements (SRS [3.1.2.1.2](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit) & SDD [5.5.2.2](#_mo1i1jrevon5))

Advertisements will be generated and provided to the application through the use of an API limited to AdWords API, Facebook Ads API, and/or the Amazon Advertising API.

#### 5.3.4.4 OpenWeatherMap API (SRS [3.6.4](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit) & SDD [5.5.2.7](#_6dws3alt3e2l))

OpenWeatherMap API will provide current and future weather data for the selected area.

#### 5.3.4.5 Emergency Services API (SRS [3.1.2.1.6](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit) & SDD [5.5.2.6](#_h1c6artxmcyn))

Emergency services will be provided through the use of the Emergency Numbers API and local emergency services found through the users location.

#### 5.3.4.6 Local businesses (SRS [3.1.2.1.5](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit) & SDD [5.5.2.5](#_704vcgm8gmlh))

Local businesses will be able to advertise and directly connect with the user by sending product information and locations to the Trailru application.

The traffic and data from the external interfaces will be sent directly to the users device and handled within the application. Each API will require unique keys to interact and will be generated by the users profile after creation.

# 5.4 Logical

According to page 35, section 5.4 of the IEEE standard, “The purpose of the Logical viewpoint is to elaborate existing and designed types and their implementations as classes and interfaces with their structural static relationships. This viewpoint also uses examples of instances of types in outlining design ideas.”

This section contains UML class diagrams that describe the different objects we will be using in Trailru. It describes the Trail, User, Review, Comment, Map, and Pinpoint classes. In addition to each of the UML class diagrams, there are descriptions for each of the properties and methods in each of those classes. These objects will be used on the client side for displaying trail and user information. These classes will be interfacing with the Trailru database server outlined in [Section 5.3](#_wijsko5d4p8s) to retrieve the relevant data.

## 5.4.1 Trail Class ([SRS 1.2.2-1.2.8](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [3.2.2](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

|  |
| --- |
|  |
| *Figure 5.4.1: Trail Class UML* |

#### 5.4.1.1 Trail Name

Stores the trail name as a string. This is the text that will define which trail we are currently viewing or working with.

#### 5.4.1.2 Trail ID

Stores the trail id as an integer. This is the variable used to reference a specific trail other than accessing it by a string variable known as trail name.

#### 5.4.1.3 Difficulty

Stores the difficulty level as an integer. This will provide a hiker with information about the difficulty level a specific trail is rated.

#### 5.4.1.4 Distance

Stores the distance of a trail in miles as a float. Distance is the variable associated with the length of the hike in miles, and will inform the user how far they will have to travel on the trail.

#### 5.4.1.5 Age Range

Stores the age range as a string. This will be information displayed about the age range that is suitable for a specific trail.

#### 5.4.1.6 Pets

Stores a boolean to determine whether pets are allowed or not. If the variable pets is true, the trail will display that it is pet friendly.

#### 5.4.1.7 Trail Type

Stores the type of trail as a string.

#### 5.4.1.8 Description

Stores a description of the trail as a string. This variable will help describe the trails landscape and key features of the hike.

#### 5.4.1.9 Get Trail Name

Returns the name of the trail, and will return the trailName variable.

#### 5.4.1.9 Set Trail Name

Sets the member variable trailName equal to a string passed as a parameter.

#### 5.4.1.10 Get Trail ID

Returns the id of the trail, and will return the trail id variable.

#### 5.4.1.10 Set Trail ID

Sets the trail id member variable equal to an integer parameter passed into the function.

#### 5.4.1.11 Get Difficulty

Returns the difficulty level of the trail, and returns the difficulty variable as an integer.

#### 5.4.1.11 Set Difficulty

Sets the difficulty member variable equal to an integer parameter passed in the set difficulty function.

#### 5.4.1.12 Get Distance

Returns the distance of the trail, and returns the distance member variable as a float.

#### 5.4.1.12 Set Distance

Sets the distance member variable equal to a float value passed into the set distance member function.

#### 5.4.1.13 Get Age Range

Returns the age range, and returns the member variable.

#### 5.4.1.13 Set Age Range

Sets the age range variable equal to the string passed into the set age range member function.

#### 5.4.1.14 Pets Allowed

Returns the pets variable that contains a boolean. This member function will return true or false to determine whether or not pets are allowed on the trail.

#### 5.4.1.14 Set Pets Allowed

Sets the pets member variable equal to the boolean passed as a parameter into the set pets allowed function.

#### 5.4.1.15 Get Trail Type

Returns the trail type variable, and returns the string.

#### 5.4.1.15 Set Trail Type

Sets the trail type string member variable equal to the string passed into the set trail type member function.

#### 5.4.1.16 Get Description

Returns the description of the trail, and returns the description variable.

#### 5.4.1.16 Set Description

Sets the description of the trail equal to the string passed into the set description member function.

#### 5.4.1.17 Add Trail Photo

Relies on database schema to add a new photograph by sending a reference to the photo location in a directory to the database.

#### 5.4.1.17 Get Trail Photo

Requires a photo id from the database schema. The function will request the photo url, and return the url of the photo.

## 5.4.2 User Class ([SRS 3.2.6](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

|  |
| --- |
|  |
| *Figure 5.4.2: User Class UML* |

#### 5.4.2.1 User Id

Stores the user id as an int of the user account.

#### 5.4.2.2 Username

Stores the username as a string of the user account.

#### 5.4.2.3 Token

Stores the token as a string. A hashed token is used like a password that allows the mobile application to communicate with the server so it can keep the credential.

#### 5.4.2.4 Email

Stores the user’s email as a string.

#### 5.4.2.5 Edit User

Allows the users to edit their account information.

#### 5.4.2.6 Delete User

Allows the users to delete their user account.

#### 5.4.2.7 Get Username

Gets the username from the user and returns username.

#### 5.4.2.8 Set Username

Set the username variable to the one passed in.

5.4.2.9 Hash Password

Hashes the password and returns a hashed string password

#### 5.4.2.10 Get Token

Gets the password from the user, hashes it, and then sends the hashed password to the server. The server returns a token to be used in requests.

#### 5.4.2.11 Set Token

Once the server returns the token, it saves the token to the account as a parameter.

## 5.4.3 Feedback Class ([SRS 3.4.3, SRS 3.2.3](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

The Feedback class is a base class from which the Review and Comment classes are derived.

|  |
| --- |
|  |
| *Figure 5.4.3:* *Feedback class UML diagram* |

#### 5.4.3.1 Get Id

Returns the id variable.

#### 5.4.3.2 Set Id

Sets the id variable to the one passed in.

#### 5.4.3.3 Get Trail Id

Returns the trail id variable.

#### 5.4.3.4 Set Trail Id

Sets the trail id variable to the one passed in.

#### 5.4.3.5 Set User Id

Sets the user id variable to the one passed in.

#### 5.4.3.6 Get User Id

Returns the user id variable.

#### 5.4.3.7 Get Body

Returns the body variable.

#### 5.4.3.8 Set Body

Stores the body. This is the main text.

The following two classes (Review and Comment) are derived classes from the Feedback base class.

### 5.4.3.1 Review Class ([SRS 3.4.3](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

|  |
| --- |
|  |
| *Figure 5.4.3.1: Review Class UML* |

#### 

#### 5.4.3.1.1 Add Review

Adds the review that was passed into the database.

#### 5.4.3.1.2 Delete Review

Deletes a review from the database based on the passed in review id.

#### 5.4.3.1.3 Edit Review

Edits a review from the database based on the passed in id and passed in new review information.

#### 5.4.3.1.4 Get Title

Returns the title variable.

#### 5.4.3.1.5 Set Title

Sets the title variable to the one passed in.

#### 5.4.3.1.6 Get Rating

Returns the rating variable.

#### 5.4.3.1.7 Set Rating

Sets the rating variable to the one passed in.

### 5.4.3.2 Comment Class ([SRS 3.2.3](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

|  |
| --- |
|  |
| *Figure* *5.4.3.2: Comment class UML* |

#### 

#### 5.4.3.2.1 Add Comment

Displays a comment supplied by the user.

#### 5.4.3.2.2 Edit Comment

Writes over an existing comment data with new data supplied by the user.

#### 5.4.3.2.3 Delete Comment

Deletes the comment from the database.

#### 5.4.3.2.4 Get Parent Id

Returns a unique number associated with the Parent Id.

#### 5.4.3.2.5 Set Parent Id

Sets a unique number associated with the Parent Id.

## 5.4.5 Map Class ([SRS 1.2.8, 3.2.1](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [3.6.4](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

|  |
| --- |
|  |
| *Figure 5.4.5: Map Class UML* |

#### 5.4.5.1 Longitude

Stores the current longitude of a given user.

#### 5.4.5.2 Latitude

Stores the current latitude of a given user.

#### 5.4.5.3 Pin Points

Stores a list of pin points that the user added.

#### 5.4.5.4 Terrain Description

Stores the description of the terrain at a given area.

#### 5.4.5.5 Terrain Difficulty

Stores the difficulty of the terrain at a given area.

#### 5.4.5.6 Get Longitude

Returns the longitude.

#### 5.4.5.7 Set Longitude

Sets the longitude variable to the one passed in.

#### 5.4.5.8 Get Latitude

Returns the latitude.

#### 5.4.5.9 Set Latitude

Sets the latitude variable to the one passed in.

#### 5.4.5.10 Get Pinpoints

Returns all the pinpoints the user set on a given map.

#### 5.4.5.11 Add Pinpoint

Adds a pinpoint to the pinpoints list.

#### 5.4.5.12 Delete Pinpoint

Deletes a pinpoint from the pinpoints list according to the passed index.

#### 5.4.5.13 Get Terrain Description

Returns the terrain description.

#### 5.4.5.14 Set Terrain Description

Sets the terrain description variable to the one passed in.

#### 5.4.5.15 Get Terrain Difficulty

Returns the terrain difficulty.

#### 5.4.5.16 Set Terrain Difficulty

Sets the terrain difficulty variable to the one passed in.

## 5.4.6 Pinpoint Class ([SRS 1.2.8, 3.2.1](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

|  |
| --- |
|  |
| *Figure 5.4.6: Pinpoint Class UML* |

#### 5.4.6.1 Longitude

Stores the longitude of a given pinpoint.

#### 5.4.6.2 Latitude

Stores the latitude of a given pinpoint.

#### 5.4.6.3 Color

Stores the color of a given pinpoint.

#### 5.4.6.4 Id

Stores the id of a given pinpoint.

#### 5.4.6.5 Get Longitude

Returns the longitude of the pinpoint

#### 5.4.6.6 Set Longitude

Sets the longitude variable to the one passed in.

#### 5.4.6.7 Get Latitude

Returns the latitude of the pinpoint

#### 5.4.6.8 Set Latitude

Sets the latitude variable to the one passed in.

#### 5.4.6.9 Get Color

Returns the color of the pinpoint.

#### 5.4.6.10 Set Color

Sets the color variable to the one passed in.

#### 5.4.6.11 Get Id

Returns the id of the pinpoint.

#### 5.4.6.12 Set Id

Sets the id variable to the one passed in.

## 5.4.7 Trail List Class

|  |
| --- |
|  |
| *Figure 5.4.7: Trail List Class UML* |

#### 5.4.7.1 Trail List

Stores the list of trails

#### 5.4.7.2 Trail

Stores the trail as a member of type Trail

#### 5.4.7.3 Trail List Id

Stores the id of a trail list as an integer

#### 5.4.7.4 Get Trail List

Returns the trail list

#### 5.4.7.5 Set Trail List

Sets the trail list to the list passed in.

#### 5.4.7.6 Get Trail

Returns the trail variable.

#### 5.4.7.7 Set Trail

Sets the trail variable to the parameter passed in.

#### 5.4.7.8 Remove Trail

Removes the trail specified by the trail list id passed in.

#### 5.4.7.9 Add Trail

Adds a trail passed in as a parameter to the trail list.

#### 5.4.7.10 Get Trail List Id

Returns the trail list id variable.

#### 5.4.7.11 Set Trail List Id

Sets the trail list id variable equal to the parameter passed in.

# 5.5 Dependency

According to IEEE 1016-2009 section 5.5 on page 17, the dependency viewpoint specifies the relationship of interconnection and access among entities such as shared information, order of execution, and parameterization of interfaces.

## 5.5.1 UML Component Entities

The number of UML Component Entities refers to the number of API’s that Trailru will be utilizing. The following subsections will describe each entity as well as their relationship with another entity if they have one. The entities that Trailru uses are as follows:

5.5.1.1 Camping API

5.5.1.2 Advertising API

5.5.1.3 Mapping API

5.5.1.4 Trairu REST API

5.5.1.5 Local Business API

5.5.1.6 Safety API

5.5.1.7 Weather API

## 5.5.2 UML Component Diagrams

#### 5.5.2.1 Camping Information API

|  |  |  |
| --- | --- | --- |
| Type | Purpose | Description |
| API | To provide the user with the most accurate and up to date camping information. | For the most accurate camping information to be made available, the Trailru application will access the API’s from various camping websites, Recreation.gov, and the National Parks website. |

Related SDD Sections: [5.4.1](#g0vzzustxxxz)

Design Concerns from SRS: [3.1.2.1](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.p19ireicog8), [3.4.2](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.blz5w41bpzip), [3.2.2](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.k0qnb9pqmfy9)



Figure 5.5.2.1.1 Component Diagram – Camping Information API

#### 5.5.2.2 Advertising API

|  |  |  |
| --- | --- | --- |
| Type | Purpose | Description |
| API | Allows for all possible advertisements on the Trailru application to be accessible. | For Trailru to make any possible advertisements that are desired to be present on the application, there must be a way for the application to access these advertisements as well as for the users to access them from the application. Both of these functionalities will be achieved through the various API’s from the various potential companies with desired advertisements. |

Related SDD Sections: None

Design Concerns from SRS: [3.1.2.1.2](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.240i158byuki)

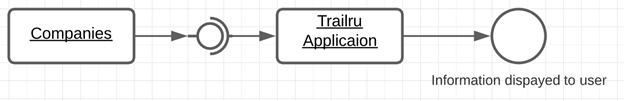


Figure 5.5.2.1.2 Component Diagram – Advertising API

#### 5.5.2.3 Maps API

|  |  |  |
| --- | --- | --- |
| Type | Purpose | Description |
| API | Allows for the downloading of the map for the specified hiking trail. | For Trailru to allow access of the map of any specified hiking trail to the user, the use of API’s is needed. Specifically, Trailru will utilize the API’s from Google Maps as well as the affiliate companies of Google Maps. |

Related SDD Sections: [5.4.1](#g0vzzustxxxz), [5.4.5](#u7mi2g2mmmsk), [5.4.6](#j96zu0jiue4a)

Design Concerns from SRS: [3.1.2.1.3](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.28dt37m0dw5e), [3.4.1](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.6l8vtxy7dn4y), [3.2.1](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.r601wdqhub5y), [4.6.4](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.rpm848d6lvmj)

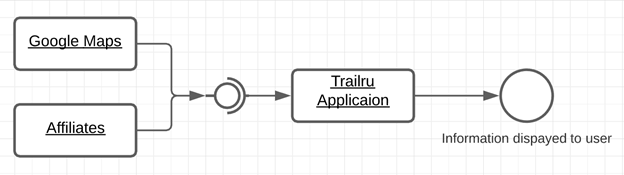


Figure 5.5.2.1.3 Component Diagram – Map API

#### 5.5.2.4 Trailru REST API

|  |  |  |
| --- | --- | --- |
| Type | Purpose | Description |
| API | Provide the user with the ability to save and receive data within the application. | For Trailru to enable the user to save and receive data within the application, the use of internal APIs will be used. |

Related SDD Sections: [5.2.2](#po6taq4qglxp), [5.2.4](#qh562llyx5hp), [5.2.8](#6ktzsx7bora6), [5.2.9](#g4m80hjz9si4), [5.2.17](#92z0rqenox8v), [5.2.18](#prxt3lhkdizq), [5.4.1](#g0vzzustxxxz), [5.4.2](#beq5fk4zxhzn), [5.4.3](#ufkpzw4r9oxd), [5.4.4](#4hopuznzhcxz)

Design Concerns from SRS: [3.1.2.1.4](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.totjl9a1667i), [3.4.5](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.3ly7bpmybxdi), [3.2.6](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.ylc18pkz9rh2), [4.5.1](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.kl8zibcuw1gc)

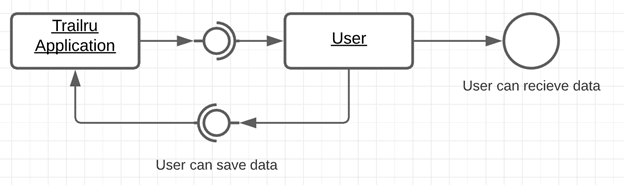


Figure 5.5.2.1.4 Component Diagram – Trailru API

#### 5.5.2.5 Local Businesses API

|  |  |  |
| --- | --- | --- |
| Type | Purpose | Description |
| API | Provides the user with a list of nearby businesses they can visit to get the supplies they will need for their hike. | For Trailru to generate a list of all the nearby businesses that will benefit the user, the application will need to use the API’s from these businesses. Specifically, API’s that identify the type of supplies are sold and where that business is in relation to the user. Meaning, that the use of the Google Maps API will also be needed. |

Related SDD Sections: [5.2.25](#hu9wk65fw59a), [5.4.5](#u7mi2g2mmmsk), [5.4.6](#j96zu0jiue4a)

Design Concerns from SRS: [3.1.2.1.3](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.9296wqn5x249), [3.1.2.1.5](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.9296wqn5x249), [3.4.7](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.hbz0eagytnbr), [3.2.8](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.2ww2nqw7j2qi)

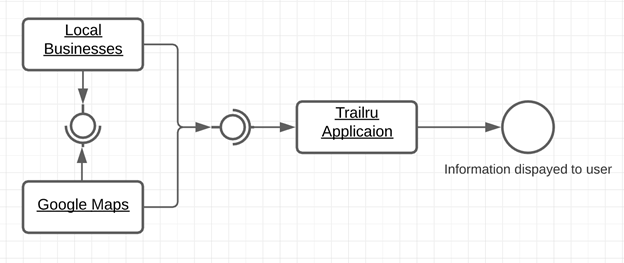


Figure 5.5.2.1.5 Component Diagram – Local Businesses API

#### 5.5.2.6 Safety API

|  |  |  |
| --- | --- | --- |
| Type | Purpose | Description |
| API | Connects Trailru to local emergency services to ensure the safety of the users. | For Trailru to ensure the safety of its users, there needs to be a way for the application to be connected to emergency services to provide assistance to any user who may need it. This connection will be done through a series of API’s connected to these emergency services websites to collect information such as phone numbers, locations, services provided, etc… |

Related SDD Sections: [5.2.15](#wqa8zuz6t5dj), [5.2.22](#searfbue6lm3), [5.2.29](#vvqavlrdhp83)

Design Concerns from SRS: [3.1.2.1.6](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.i2s3xags24ay), [3.4.4](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.kzghzs25n1st), [3.2.4](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.v51tbgx7rlv)

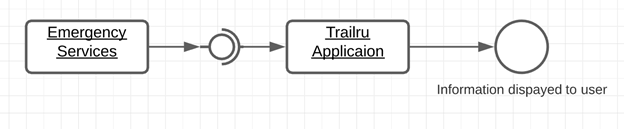


Figure 5.5.2.1.6 Component Diagram – Safety API

#### 5.5.2.7 Weather API

|  |  |  |
| --- | --- | --- |
| Type | Purpose | Description |
| API | Provide the user with the ability to view current and accurate weather data. | For the most accurate weather data to be readily available to the user, then the use of API’s will be needed. Specifically, Trailru will be connected to the Weather Channel’s website API to receive the most accurate and up to date weather information. |

Related SDD Sections: [5.2.33](#ozn9dyr2jkqq), [5.2.34](#5o2illz9axf9)

Design Concerns from SRS: [3.1.2.1.7](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.cfuj7ojhu4gn), [3.4.4](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.kzghzs25n1st), [3.2.4](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.v51tbgx7rlv)

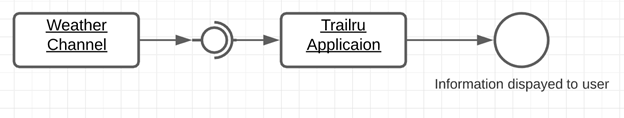


Figure 5.5.2.1.7 Component Diagram – Weather API

# 5.6 Information

This section will be seen using the **Information Viewpoint**. According to the IEEE standard 1016-2009 Page 17 Section 5.6, this viewpoint is used when there is a substantial amount of data that is persistent in the application, which is true for the needs of this design, and specially for a mobile application, This requires the application to process and keep information in offline mode, since it will be the mode in which will be used in most circumstances of enjoyment of trails.

This viewpoint contains and defines the persistent information in Trailru. The information viewpoint contains concerns regarding entities such as: persistent information, database, content, and access.

### 

### 5.6.1 Local Application/Client Database

Trailru shall feature both **synchronous** and **asynchronous** design.

#### 5.6.1.1 Local Database over SQLite

To achieve this functionality, it will have local storage in the mobile application, to maintain information across sessions and allow the user to operate in an asynchronous way in Offline mode.

The chosen database engine is SQLite for its ACID compliance features which allows for complete and consistent transactions of information persistence and retrieval. It's open source will work effectively and locally in both iOS and Android targeted platforms.

#### 5.6.1.2 Local Database Information Encryption

SQLite features complete encryption of information using the SQLCipher extension which allows for 256 bits AES encryption. It uses OpenSSL libcrypto for the specific implementation, allowing the application to protect sensitive information like user personal information, GPS location, planned trails, and friends and contacts details.

### 5.6.1.3 Local Data Layer Diagram

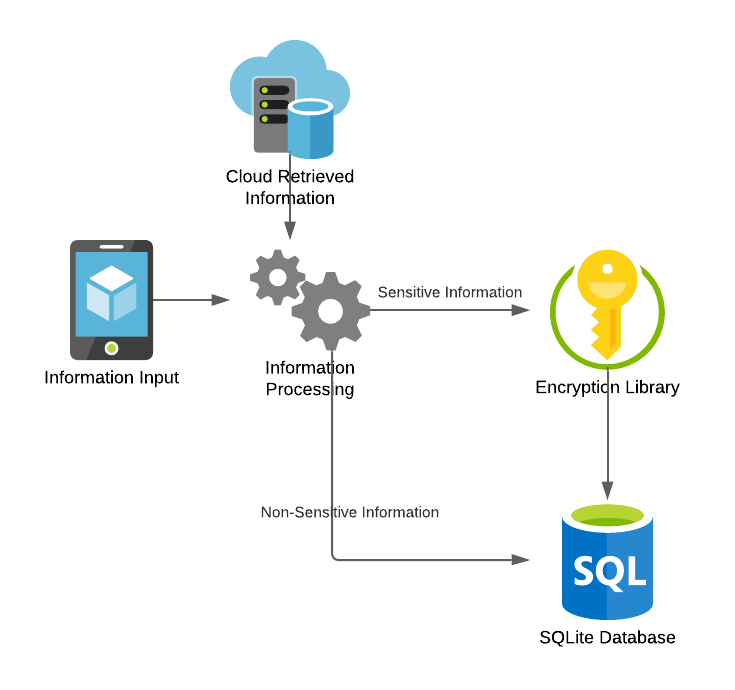


Figure Information 5.6.1.3

### 5.6.2 Back End / Server Side Database

The server side persistence layer will be reached through the server API and after business logic operations information will be stored in the chosen SQL database.

#### 5.6.2.1 Server Side Database over MySQL 8.0+

After careful review and consideration, we have chosen MySQL 8.0+ as the SQL database for our server side persistence layer thanks to its performance, ease of use, robustness and open source license.

We understand more advanced and featured implementations of SQL databases exist, but since the main focus of the Trailru application is the client app, we believe MySQL 8.0 meets our needs, and highlight some special features that allow for easier deployment and performance. Of those the main reasons are:

1. **Open Source License:** Reducing costs to the server side implementation, and allowing for extensions to be built if needed.
2. **High performance on reading transactions**, since those are the ones to operate the most in TrailRu, we expect to have much more reading of information from the server, than actual uploading or writing, since the majority of users will tend more to retrieve information about different trails than provide it.
3. **Low server side memory footprint**, other implementations of SQL databases, create an operating system process for each different connection, but MySQL 8 creates a thread, which is much faster to create and has lower memory footprint (Threads per connection ~250KB vs Process per connection ~8MB )

Encryption will be achieved through File System encryption at the Operating System level, and connections to the database will be only authorized from the application server or API and encrypted over SSL.

The specific version 8 is very important for the sake of default SSL encrypted connections and performance over the previous version (version 5.7).

#### 5.6.2.2 Database Entities definition

Having created the “trailru” database through command interface:

|  |
| --- |
| create database trailru; use trailru; |

The following database entities are defined in SQL DDL (Database Description Language):

|  |
| --- |
| create table tags (  tag\_id int auto\_increment  primary key,  tag varchar(30) not null,  tag\_type varchar(40) default 'TRAIL\_FEATURE' null comment 'Potential values may include ''TRAIL\_FEATURE'', ''SAFETY'', etc.',  constraint tags\_tag\_uindex  unique (tag) );  create table trails (  trail\_id int auto\_increment  primary key,  trail\_name varchar(90) not null,  trail\_description text null,  overall\_rating float(4,2) null,  traffic\_score tinyint default 0 null,  created\_at timestamp default CURRENT\_TIMESTAMP not null,  updated\_at timestamp default CURRENT\_TIMESTAMP not null on update CURRENT\_TIMESTAMP );  create table trail\_markers (  trail\_marker\_id int auto\_increment  primary key,  trail\_id int not null,  marker\_name varchar(90) not null,  marker\_description text null,  coordinates point not null,  marker\_type varchar(50) default 'POINT\_OF\_INTEREST' null comment 'Potential values include TRAIL\_HEAD, POINT\_OF\_INTEREST, DANGER',  marker\_source varchar(30) default 'USER\_SUBMITTED' null,  created\_at timestamp default CURRENT\_TIMESTAMP not null,  updated\_at timestamp default CURRENT\_TIMESTAMP null on update CURRENT\_TIMESTAMP,  constraint trail\_markers\_trails\_trail\_id\_fk  foreign key (trail\_id) references trails (trail\_id)  on update cascade on delete cascade );  create table trail\_meta (  trail\_meta\_id int auto\_increment  primary key,  trail\_id int not null,  meta\_key varchar(120) not null,  meta\_value text null,  created\_at timestamp default CURRENT\_TIMESTAMP not null,  constraint trail\_meta\_trails\_trail\_id\_fk  foreign key (trail\_id) references trails (trail\_id)  on update cascade on delete cascade );  create table trail\_tags (  trail\_id int not null,  tag\_id int not null,  constraint trail\_tags\_tags\_tag\_id\_fk  foreign key (tag\_id) references tags (tag\_id)  on update cascade on delete cascade,  constraint trail\_tags\_trails\_trail\_id\_fk  foreign key (trail\_id) references trails (trail\_id)  on update cascade on delete cascade );  create table trail\_traffic (  trail\_traffic\_id int auto\_increment  primary key,  trail\_id int not null,  traffic\_score tinyint default 0 not null,  created\_at timestamp default CURRENT\_TIMESTAMP not null,  constraint trail\_traffic\_trails\_trail\_id\_fk  foreign key (trail\_id) references trails (trail\_id)  on update cascade on delete cascade );  create table trail\_weather (  trail\_weather\_id int auto\_increment  primary key,  trail\_id int not null,  temperature float(6,2) not null,  weather\_code varchar(40) default 'CLEAR' not null comment 'Potential values include ''CLEAR'', ''RAIN'', ''SNOW'', ''CLOUDY'', etc.',  created\_at timestamp default CURRENT\_TIMESTAMP not null,  constraint trail\_weather\_trails\_trail\_id\_fk  foreign key (trail\_id) references trails (trail\_id)  on update cascade on delete cascade );  create table users (  user\_id int auto\_increment  primary key,  email varchar(60) not null,  first\_name varchar(40) null,  last\_name varchar(40) null,  created\_at timestamp default CURRENT\_TIMESTAMP not null,  updated\_at timestamp default CURRENT\_TIMESTAMP not null on update CURRENT\_TIMESTAMP,  constraint users\_email\_uindex  unique (email) );  create table trail\_photos (  trail\_photo\_id int auto\_increment  primary key,  trail\_id int not null,  user\_id int not null,  photo\_url varchar(120) not null,  created\_at timestamp default CURRENT\_TIMESTAMP not null,  constraint trail\_photos\_trails\_trail\_id\_fk  foreign key (trail\_id) references trails (trail\_id)  on update cascade on delete cascade,  constraint trail\_photos\_users\_user\_id\_fk  foreign key (user\_id) references users (user\_id)  on update cascade on delete cascade );  create table trail\_reviews (  rating\_id int auto\_increment  primary key,  trail\_id int not null,  user\_id int not null,  review text null,  rating float(4,2) default 5.00 not null,  created\_at timestamp default CURRENT\_TIMESTAMP not null,  updated\_at timestamp default CURRENT\_TIMESTAMP null on update CURRENT\_TIMESTAMP,  constraint trail\_reviews\_trails\_trail\_id\_fk  foreign key (trail\_id) references trails (trail\_id)  on update cascade on delete cascade,  constraint trail\_reviews\_users\_user\_id\_fk  foreign key (user\_id) references users (user\_id)  on update cascade on delete cascade );  create table user\_trail\_favorites (  user\_trail\_favorite\_id int auto\_increment  primary key,  trail\_id int not null,  user\_id int not null,  created\_at timestamp default CURRENT\_TIMESTAMP not null,  constraint user\_trail\_favorites\_trails\_trail\_id\_fk  foreign key (trail\_id) references trails (trail\_id)  on update cascade on delete cascade,  constraint user\_trail\_favorites\_users\_user\_id\_fk  foreign key (user\_id) references users (user\_id)  on update cascade on delete cascade );  create table user\_trail\_recommendations (  user\_trail\_recommendation\_id int auto\_increment  primary key,  user\_id int not null,  trail\_id int not null,  priority\_score int default 0 not null,  created\_at timestamp default CURRENT\_TIMESTAMP null,  constraint user\_trail\_recommendations\_pk\_2  unique (user\_id, trail\_id),  constraint user\_trail\_recommendations\_trails\_trail\_id\_fk  foreign key (trail\_id) references trails (trail\_id)  on update cascade on delete cascade,  constraint user\_trail\_recommendations\_users\_user\_id\_fk  foreign key (user\_id) references users (user\_id)  on update cascade on delete cascade ); |

#### 5.6.2.3 Database Entities Relationship Diagram

Having created the database entities, the following will be the corresponding diagram:



### 5.6.3 GPS

#### 5.6.3.1 GPS functionality

GPS functionality will be asynchronous allowing the user to have the GPS running in the background while managing other apps during planning or hiking.

The Brief Mapping of Trailru GPS Function

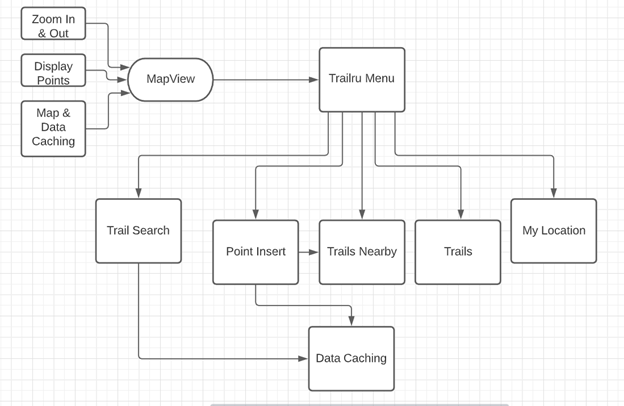
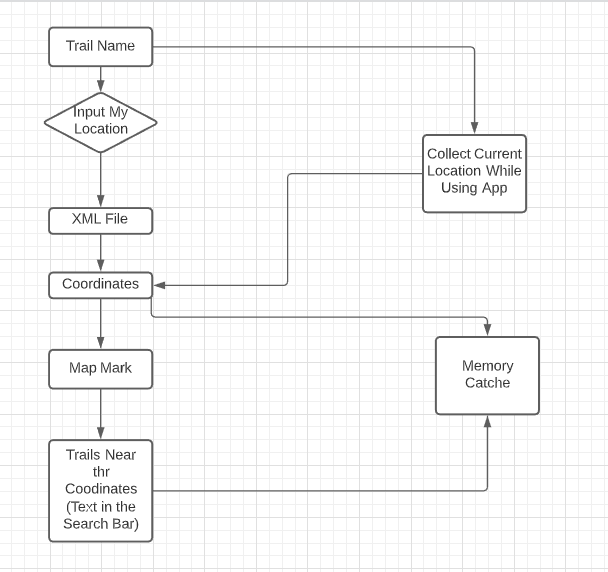


Figure Information 5.6.2.1

Above is a brief outline of the general stages of GPS mapping system in Trailru mobile application, this section will focus on the implementation of GPS function design. The application is composed by the Google maps interface.

Near Hiking Trail

5.6.2.2

5.6.3.2 The process of the function in figure is to show users the near hiking trail spots around them. This design pattern is mainly focused on understanding the needs of users, by designing the application based on the location spot and filter the distance of trails.

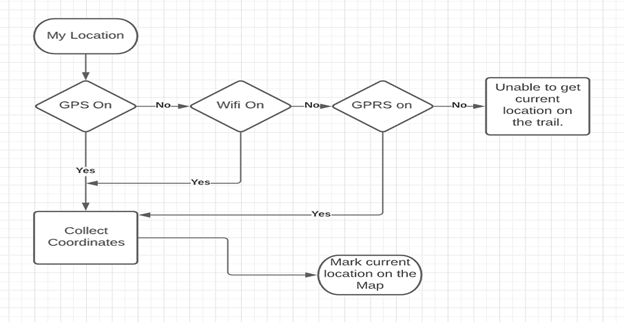
My Location (Track on the Trail)

Figure Information 5.6.3.3

#### 5.6.3.3 My Location Figure

The figure above shows how the application shall receive the current circumstances to decide if GPS is on or not. If the GPS is off, it will check the Wi-Fi and GPRS. If any of the methods is available, users will be able to use devices to collect current coordinates and display their location with icons on the trail map.

## 5.6.4 Map and Data Caching

#### 5.6.4.1 Cached Data

While using SQLite, the application can cache data needed in cell phone memory. SQLite is an open source relational database that supports standard databases and takes up a small amount of memory while the application is running.

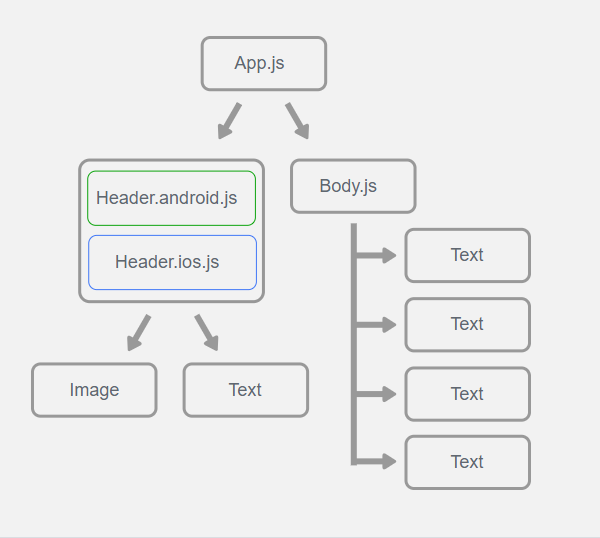
# 5.7 Patterns

This section focuses on the **patterns framework** according to the IEEE standard 1016-2009 Page 17. Section 5.7 will be concerned with the React Native framework which allows seamless cross-platform (android, ios) development.

### 5.7.1 React Native Cross-Platform Communication

The React Native framework allows for seamless cross-platform mobile development. React Naitive apps are written using React, a JavaScript framework, and compiled to platform specific code (for either IOS, Android, or the web) so that it can be written which is shared on all platforms. For example the React Naitive <TextInput> component maps both to IOS’s <UITextField> component and Android’s <EditText> component.

As stated above React Native is based on React which uses a combination of JavaScript and JSX (JavaScript XML) syntax to build user interfaces. Functions and classes are built using standard JavaScript syntax, within these functions and classes there can be code written with JSX syntax which will be rendered in the app (the <TextInput> component mentioned above is an example of a JSX tag), JSX can also have JavaScript code embedded within it. One function from the React Naitive code is used as the starting point for the whole program (much like the main function in C).



### 5.7.2 Model View Controller Design Pattern

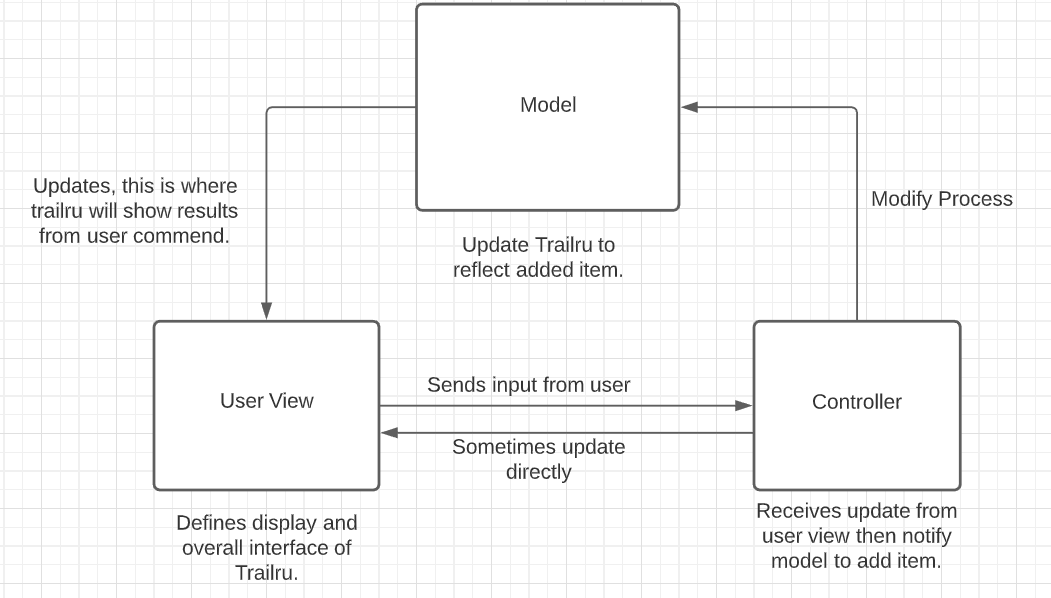
Model view controller is a software design pattern that is commonly used for user interface development process. Model view controller design pattern divides the related topics into three design components. The idea is for users to provide a command prompt to Trailru, and the modules will provide feedback and results to users. Below are the description of three components:

Model-The central component of design pattern, it directly manages the data and rules of Trailru.

User View-Representation of information such as user interface.

Controller-Inputs and converts to commend for for model or user view.

### 5.7.2.1 Diagram



# 5.8 Interface

The Interface Viewpoint is described in IEEE Std 1016-2009 on pages 19-20, section 5.8.

The user interface for Trailru will be created following the requirements in the [SRS](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.vamxzx6vuf0b). Each page will feature picture labels like home, back, and options to make the user feel more at ease and in charge. The user interface will be clearly and thoroughly labeled so the user doesn’t have to search around for the right button.

The following images are examples of the user interface.

|  |
| --- |
|  |
| *Figure* *5.8.1: User Interface Mock-up* |

### 5.8.1 Home Page

#### 5.8.1.1 Keyword Search ([SRS 1.2.1](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.q7bnhq21pq39)) A display that has a list of popular keywords that describe hikes.

|  |
| --- |
|  |
| *Figure* *5.8.1.1.1: Search box* |

#### 5.8.1.1.2 Hike Recommendations ([SRS 3.2.7](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.us2lpu1bj4cj))

A display that shows users hikes that are recommended to them.

|  |
| --- |
|  |
| *Figure* *5.8.1.2.1: Hike Recommendations* |

#### 5.8.1.3 Favorites Button ([SRS 3.2.6.1](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.us2lpu1bj4cj))

##### 5.8.1.3.1 Save as Favorite

A button that allows users to save a hike to the favorites list.

|  |
| --- |
|  |
| *Figure* *5.8.1.3.1.1: Save as Favorite Button* |

##### 5.8.1.3.2 Favorites Button

A button allows users to access hikes that they have previously saved as a favorite, signified by a heart icon.

|  |
| --- |
|  |
| *Figure* *5.8.1.3.2.1: Favorites Button* |

#### 5.8.1.4 Hike Card ([SRS 3.2.2](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.us2lpu1bj4cj))

A display that shows a summary of the information found on the hike page.

|  |
| --- |
|  |
| *Figure* *5.8.1.4.1: Hike Card* |

### 5.8.2 Hike Page

#### 5.8.2.1 Hike Photos

A display that shows photos of the hike. Tapping on the picture of the hike will let the user scroll through a selection of photos of that hike.

|  |
| --- |
|  |
| *Figure* *5.8.2.1.1: Hike Photos* |

#### 5.8.2.2 Hike Difficulty

An interface that shows the difficulty and distance of the hike. The information will appear in the upper right hand corner of the screen when a hike is selected for the user.

|  |
| --- |
|  |
| *Figure* *5.8.2.2.1: Hike Difficulty* |

#### 5.8.2.3 Hiker Traffic

Displays how busy a current trail is or how busy it is projected to be. It will be displayed directly under the photos of the hike on the left hand side. There is a circle that will be green, medium, or red to signify low, medium, or hike traffic respectively.

|  |
| --- |
|  |
| *Figure* *5.8.2.3.1: Hike Traffic* |

#### 5.8.2.4 Hike Classification

Describes what kind of hike it is and gives information about it. The description will be located under the name of the hike on the hike card.

|  |
| --- |
|  |
| *Figure* *5.8.2.4.1: Hike Classification* |

#### 5.8.2.5 Download Map [(SRS 3.4.1)](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.6l8vtxy7dn4y)

A button that allows the user to download the map for offline navigation. This button will be placed on the hike page at the bottom left of the screen. When downloading, the app will access the map API and place the downloaded map into the app titled as “Trailru\_’location’\_offlineMap”. For example: Trailru\_PassMountainTrail\_offlineMap.

|  |
| --- |
|  |
| *Figure* *5.8.2.5.1: Download Map* |

#### 5.8.2.6 Flag Inappropriate Content [(SRS 3.4.3)](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.mgcitpzcjj54)

A button that allows users to report content. This button will be located at the bottom of each review. The button will flag the review to be investigated by the Trailru team. Any review found to be inappropriate will be removed and the writer will be sent a message on their app explaining why the review was removed.

|  |
| --- |
|  |
| *Figure* *5.8.2.6.1: Flag Inappropriate Content* |

#### 5.8.2.7 Weather Information [(SRS 3.6.4)](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.p6qjt66sn6hx)

A small display showing the current weather forecast for the area of the hike. This link will be in the top left corner of the page, it will show a symbol representing the general weather. A sun, a sun with cloud cover, a raining cloud, or a snowing cloud. It will also have the current degrees in Fahrenheit.

|  |
| --- |
|  |
| *Figure* *5.8.2.7.1: Weather Information* |

### 5.8.3 Camera Page

#### 5.8.3.1 Camera View ([SRS 1.2.14](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/))

A view that shows the input from one of the device’s cameras on the screen.

#### 5.8.3.2 Switch Camera Button

A button that allows the user to switch between the front and rear cameras.

#### 5.8.3.3 Take Picture Button

A button that will take a picture and save it to the device’s memory.

|  |
| --- |
|  |
| *Figure* *5.8.3.1: Camera View* |

### 5.8.4 Write Review Page

#### 5.8.4.1 Review panel ([SRS 3.2.3](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.gfo663z1bbuj))

In the Review panel, the user is able to register his/her opinions and experiences on the trail. The Review panel contains two areas, one for text and one for images, the user can write and enhance the review with images if he/she wants.

|  |
| --- |
|  |
| *Figure* *5.8.4.1.1: Review Panel* |

#### 5.8.4.2 Submit Review Button ([SRS 3.2.3](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.gfo663z1bbuj))

Once the user finishes writing the review, he/she presses the “submit” button to save the review to be visualized by all other users who access that trail.

# 5.9 Structure

The Structure viewpoint is described in IEEE on page 20, section 5.9.The purpose of the Structure viewpoint is to describe patterns of inheritance, aggregation, or composition among software classes.

## 5.9.1 Trail Class ([SRS 1.2.2-1.2.8](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [3.2.2](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

See section [SDD 5.4.1](#_glzkiuvmk6j2) for more details on the Trail Class.

|  |
| --- |
|  |
| *Figure 5.9.1: Trail Entity Relationship Diagram* |

One Trail can have 0 to multiple Reviews. A User can have 0 to multiple Trails stored as a favorite. A Trail can have 0 to multiple Comments. A Trail has 1 aggregate of a Map. One item to note is that the Account class is in relation to the Trail List class.

## 5.9.2 User Class ([SRS 3.2.6](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

See section [SDD 5.4.2](#_c0z5h2kk7l6f) for more details on the User Class.

|  |
| --- |
|  |
| *Figure 5.9.2: User Entity Relationship Diagram* |

One user can have zero to many TrailLists. A user can also have zero to many Trails.

## 5.9.3 Review Class ([SRS 1.2.12, 3.4.3](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

See section [SDD 5.4.3](#_9hurh09h2t68) for more details on the Review Class.

|  |
| --- |
|  |
| *Figure 5.9.3: Review Entity Relationship Diagram* |

One Trail can have zero to multiple Reviews. Review is derived from Feedback.

## 5.9.4 Comment Class ([SRS 3.2.3](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

See section [SDD 5.4.4](#_gt1gjcocorm4) for more details on the Comment Class.

|  |
| --- |
|  |
| *Figure 5.9.4: Comment Entity Relationship Diagram* |

One Trail can have zero to many Comments. A Comment is derived from Feedback.

## 5.9.5 Map Class ([SRS 1.2.8, 3.2.1](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit), [3.6.4](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

See section [SDD 5.4.5](#_z62weatn0x55) for more details on the Pinpoint Class

|  |
| --- |
|  |
| *Figure 5.9.5: Map Entity Relationship Diagram* |

One Map belongs to one Trail and one Map can have zero or more Pinpoints.

## 5.9.6 Pinpoint Class ([SRS 1.2.8, 3.2.1](https://docs.google.com/document/u/0/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit))

See section [SDD 5.4.6](#_31mn9rripp74) for more details on the Pinpoint Class.

|  |
| --- |
|  |
| *Figure 5.9.6: Pinpoint Entity Relationship Diagram* |

One Map can have zero to multiple Pinpoints.

## 5.9.7 Trail List Class

See section [SDD 5.4.7](https://docs.google.com/document/d/17cHYcGt9fDXSllN5Q5_ylb63bthC1Ybx57uAGlmIFZ0/edit#heading=h.rgad1dapq6rk) for more details on the Trail List Class.

|  |
| --- |
|  |
| *Figure 5.9.7: Trail List Entity Relationship Diagram* |

One User can have zero to multiple TrailLists.

## 5.9.8 Feedback Class

See section [SDD 5.4.3](https://docs.google.com/document/d/17cHYcGt9fDXSllN5Q5_ylb63bthC1Ybx57uAGlmIFZ0/edit#heading=h.rgad1dapq6rk) for more details on the Feedback Base Class.

|  |
| --- |
|  |
| *Figure 5.9.8: Feedback Entity Relationship Diagram* |

Feedback is a base class. Comment and Reviews are derived from Feedback.

# 5.10 Interaction

The Interaction viewpoint is described in IEEE Std 1016-2009 pages 20-21, section 5.10. This section will be using UML Sequence Diagrams to demonstrate the interaction between several parts of the application.

## 5.10.1 User Interaction

#### 5.10.1.1 Trail Search ([SRS 3.2.5](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.t115115grde9), [SDD 5.8.1.1](#_3tig2wb5vtbj))

User submits a trail search. The search is made on a text box using text. Trailru performs a SQL sanitization in the text submitted by the user before submitting the request to the Trailru REST API. Trailru receives the results and displays them to the user.

|  |
| --- |
|  |
| *Figure 5.10.1.1.1: Trail Search* |

#### 5.10.1.2 Trail Recommendations ([SRS 3.2.7](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.va8hejdnf2z2), [SDD 5.8.1.1.2](#_vic6ufa92e3a))

Trailru saves several points of user data that is used in order to offer relevant trail recommendations. When the user starts up Trailru, 3 types of recommendation will be offered to the user:

1. Based on the user’s preferences
2. Based on the user’s last hikes
3. Based on the user’s equipment

These recommendations are all fetched from the Trailru REST API, which will build the recommendations based on the user data previously saved.

|  |
| --- |
|  |
| *Figure 5.10.1.2.1: Save User Data* |

|  |
| --- |
|  |
| *Figure 5.10.1.2.2: Request Hike Recommendations* |

#### 5.10.1.3 Add Hikes ([SRS 3.2.5.4](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.4uz3khtnyomc))

Users can submit a hike by pushing a button and filling out a form that includes information about the hike.

|  |
| --- |
|  |
| *Figure 5.10.1.3.1: Add Hikes* |

#### 5.10.1.4 Check Hiker Traffic ([SRS 3.2.2](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.k0qnb9pqmfy9))

Shows how busy a current trail is or how busy it is projected to be. Retrieves the information from Trailru REST API.

|  |
| --- |
|  |
| *Figure 5.10.1.4.1: Check Hiker Traffic* |

#### 5.10.1.5 Hike Distance and Difficulty ([SRS 3.2.3](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.du2dyurd9dyo))

Users can rate the difficulty of a hike and see the difficulty of the hike as rated by other users. The data can be sent to and retrieved from Trailru REST API.

#### 5.10.1.6 Hike Information ([SRS 3.2.2](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.k0qnb9pqmfy9))

The details of each hike, such as reviews, photos, and a description, are stored at Trailru REST API. The information can be retrieved and displayed to the user.

|  |
| --- |
|  |
| *Figure 5.10.1.6.1: Request Hike Information* |

#### 5.10.1.7 Download Map Coordinates ([SRS 3.2.1.1](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.dbhpdbfmvw9j))

A map of the hike can be downloaded so that it can be used in areas with little or no cell coverage.

|  |
| --- |
|  |
| *Figure 5.10.1.7.1: Download Map Coordinates* |

### 5.10.2 Trail Safety ([SRS 3.2.4](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.d8zsd4bs76u4))

Trail Safety encompasses functionalities related to the user safety. In order to ensure safety, Trailru requires interfaces with third party services that provide different services that fulfill Trailru’s capacity to ensure user’s safety. This section will demonstrate how Trailru and the user interact with those functionalities.

#### 5.10.2.1 Weather Information ([SRS 3.2.4.1](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.d8zsd4bs76u4))

Trailru will request for weather information and show it on the screen. It will refresh every 15 minutes, or, make a request through the API to the Weather Information Services every 15 minutes

|  |
| --- |
|  |
| *Figure 5.10.2.1.1: Request Weather Information* |

#### 5.10.2.2 Danger Information ([SRS 3.2.4.3](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.d8zsd4bs76u4))

The user will request Trailru to save danger information in the area. Trailru will get the GPS location and then save the danger and the location on Trailru’s servers using the Trailru REST API. Trailru from any other user will fetch all dangers in the area from the Trailru REST API and display on the screen.

|  |
| --- |
|  |
| *Figure 5.10.2.2.1: Register a Danger* |

|  |
| --- |
|  |
| *Figure 5.10.2.2.2: Request Dangers* |

#### 5.10.2.3 Rules and Regulations ([SRS 3.2.4.4](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.d8zsd4bs76u4), [SRS 3.2.4.6](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.d8zsd4bs76u4))

The user selects a trail and all the information about this trail, rules and regulations, are fetched from Trailru REST API and then displayed to the user. Forest Services are able to register rules and regulations about their trails.

|  |
| --- |
|  |
| *Figure 5.10.2.3.1: Register Rules* |

|  |
| --- |
|  |
| *Figure 5.10.2.3.2: Request Rules* |

# 

#### 5.10.2.4 Emergency Messages ([SRS 3.2.4.7](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.d8zsd4bs76u4))

User sends an emergency message. If the Internet is available, this request is sent to the Trailru REST API, which will send the request to all hikers in the area and capture any response. If there is a response, the message will be sent back to the user.

|  |
| --- |
|  |
| *Figure 5.10.2.4.1: Send Emergency Call* |

#### 5.10.2.5 Emergency Contacts ([SRS 3.2.4.2](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.d8zsd4bs76u4), [SRS 3.2.4.5](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#heading=h.d8zsd4bs76u4))

The user will be able to register an emergency contact number. In the event of an emergency, a message will be sent to the emergency contact containing the last known location of the user.

|  |
| --- |
|  |
| *Figure 5.10.2.5.1: Register Emergency Number* |

|  |
| --- |
|  |
| *Figure 5.10.2.5.2: Send Emergency Message* |

### 5.10.3 Device Permissions

#### 5.10.3.1 Requesting Camera Permissions

Trailru will request permission from the Device in order to use the phone camera.

|  |
| --- |
|  |
| *Figure 5.10.3.1.1: Request Camera Permissions* |

# 

# 5.11 State Dynamics

The state dynamics viewpoint is described in IEEE 1016-2009 on page 21, section 5.11. State Dynamics Views are represented using UML Activity Diagrams.

## 5.11.1 Design Entities

|  |  |  |
| --- | --- | --- |
| Home Page | [SRS 3.1.1.3](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.3mcd4t5jzunn) | [SDD 5.8.1](#1wn21cnpusxl) |
| Recommendations Page | [SRS 3.1.1.7](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.qln7zlijuk4h) | [SDD 5.2.2](#po6taq4qglxp), [SDD 5.8.1.1.2](#c1pxp0qumhj3), [SDD 5.10.1.2](#a9bgde4c9b76) |
| Favorites List | [SRS 1.2.2](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79) |  |
| Search Results Page | [SRS 1.2.1](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79), [SRS 3.2.5](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#bookmark=id.t115115grde9) | [SDD 5.2.23](#dvacas20zouv), [SDD 5.10.1.1](#43cekvv9o89r) |
| Location Page | [SRS 3.1.1.1](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#bookmark=id.pggh0wd7lifj), [SRS 3.2.1](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.1v6e4k5bt42y) |  |

### 5.11.1.1 Outline of Entities and Their Activities

This section outlines the entities that affect how the state of the application changes. It begins with the user opening the application.

1. The user opens up the application on the mobile device.
2. **Home Page:** This is the page from which the user is able to navigate to other pages within the application. This includes buttons to navigate to the recommendations page ([3](#kix.6uunxdlahm0o)) and favorites list ([4](#kix.lj7ja6kcex5b)), as well as a search bar ([6](#kix.r3njwn9incjb)) that allows the user to initialize a search and view search results ([5](#kix.9tpxk2ffuwkm)).
3. **Recommendations Page:** This page shows a list of recommended trails to the user. Each trail listed shall link to the respective location page ([8](#kix.gre18z2z75b2)) for the trail.
4. **Favorites List:** This page is where users can view all trails that they have marked as “favorite.” Each trail listed shall link to the respective location page ([8](#kix.gre18z2z75b2)).
5. **Search Results Page:** This page contains a list of hyperlinks based on the search bar ([6](#kix.r3njwn9incjb)) and filter ([7](#kix.w8lurm6gk6zh)) inputs. Each trail listed shall link to the respective location page ([8](#kix.gre18z2z75b2)).
6. **Search Bar:** This is the text entry area on the home page ([2](#kix.8c2cnbquyv72)) search results page ([5](#kix.9tpxk2ffuwkm)) that specifies what trails the user is searching for. Changing the search bar’s text shall update the information on the search results page ([5](#kix.9tpxk2ffuwkm)).
7. **Search Filter**: This is a selection area on the search results page ([5](#kix.9tpxk2ffuwkm)) where the user can narrow down their search based on the type of camping trip or hike the user is searching for. These filter options shall include the following, at minimum: day hike, drive-in camping, and backpack hike-in camping ([SRS 3.2.5.3](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#bookmark=id.t115115grde9)). Changing filter options shall update the information on the search results page ([5](#kix.9tpxk2ffuwkm)).
8. **Location Page:** This page displays information to the user on a given trail. This includes details of the trail, a map of the trail, warnings, and pictures of the trail. If the user taps on a picture, it will open the image in a lightbox viewer ([9](#kix.8kzjnh5spjhc)).
9. **Picture Lightbox:** This is a lightbox showing the selected picture, a user-submitted description, information on the user that submitted the image, and the time at which the picture was submitted. When the user exits the lightbox, the user shall be returned to the screen they were on before opening the lightbox.
10. **Home Button:** Each app page shall have a home button in the top left corner that brings the app back to the home page ([2](#kix.8c2cnbquyv72), see [SRS 3.1.1.3.1](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#bookmark=id.2k4cdlr14mfr)).

## 5.11.2 State Dynamics Views

### 5.11.2.1 Home Page

|  |
| --- |
|  |
| *Figure 5.11.2.1.1 Home Page Diagram* |

*The red circles in Figure 5.11.2.1.1 Home Page, reference* [*1*](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.xe68mmq546bm)*,* [*2*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.1rf9gpq)*,* [*3*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.4bewzdj)*,* [*4*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.2qk79lc)*,* [*5*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.15phjt5)*, and* [*6*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.3pp52gy) *from* [*5.11.1.1*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#heading=h.3c9z6hx)

*The green circles reference the items below*

1. From the Home Page ([2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.2k82xt6)), if the user presses the “Favorites” button, the app shall display the Favorites Page ([4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.3jd0qos)) to the user.
2. From the Home Page ([2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.2k82xt6)), if the user uses the search bar ([6](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.4ihyjke)), the app shall display the Search Results Page ([5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1yib0wl)) to the user.
3. From the Home Page ([2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.2k82xt6)), if the user presses the “Recommendations” button, the app shall display the Recommendations Page ([3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.zdd80z)) to the user.

### 5.11.2.2 Recommendations Page

### 

|  |
| --- |
|  |
| *Figure 5.11.2.2.1 Recommendations Page Diagram* |

*Figure 5.11.2.2.1 Recommendations Page, references* [*2*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.1rf9gpq)*,* [*3*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.4bewzdj)*,* [*8*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.jzpmwk)*, and* [*10*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.vpp6j315edbo) *from* [*5.11.1.1*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#heading=h.3c9z6hx)

*The green circles reference the items below*

1. From the Recommendations Page ([3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.zdd80z)), if the user selects a trail from the generated list, the app shall transition to the respective Location Page ([8](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1csj400)) for that trail.
2. From the Recommendations Page ([3](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.zdd80z)), if the user presses the home button ([10](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.3t3zi4wtlf68)), the app shall return to the Home Page ([2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.2k82xt6)).

### 5.11.2.3 Favorites List

### 

|  |
| --- |
|  |
| *Figure 5.11.2.3.1 Favorites List Diagram* |

*Figure 5.11.2.3.1 Favorites List, references* [*2*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.1rf9gpq)*,* [*4*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.2qk79lc)*,* [*8*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.jzpmwk)*, and* [*10*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.vpp6j315edbo) *from* [*5.11.1.1*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#heading=h.3c9z6hx)

*The green circles reference the items below*

1. From the Favorites List ([4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.3jd0qos)), if the user selects a trail from the list, the app shall transition to the respective Location Page ([8](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1csj400)) for that trail.
2. From the Favorites List ([4](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.3jd0qos)), if the user presses the home button ([10](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.3t3zi4wtlf68)), the app shall return to the Home Page ([2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.2k82xt6)).

### 5.11.2.4 Search Results

### 

|  |
| --- |
|  |
| *Figure 5.11.2.4.1 Search Results Diagram* |

*Figure 5.11.2.4.1 Search Results, references* [*2*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.1rf9gpq)*,* [*5*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.15phjt5)*,* [*6*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.3pp52gy)*,* [*7*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.24ufcor)*,* [*8*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.jzpmwk)*, and* [*10*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.vpp6j315edbo) *from* [*5.11.1.1*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#heading=h.3c9z6hx)

*The green circles reference the items below*

1. From the Search Results Page ([5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1yib0wl)), if the user updates his/her search query in the search bar ([6](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.4ihyjke)), the app shall update the results shown on the Search Results Page ([5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1yib0wl)).
2. From the Search Results Page ([5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1yib0wl)), if the user updates his/her search filter ([7](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.2xn8ts7)), the app shall update the results shown on the Search Results Page ([5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1yib0wl)).
3. From the Search Results Page ([5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1yib0wl)), if the user selects a trail from the list, the app shall transition to the respective Location Page ([8](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1csj400)) for that trail.
4. From the Search Results Page ([5](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1yib0wl)), if the user presses the home button ([10](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.3t3zi4wtlf68)), the app shall return to the Home Page ([2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.2k82xt6)).

### 5.11.2.5 Location Page

### 

|  |
| --- |
|  |
| *Figure 5.11.2.5.1 Location Page Diagram* |

*Figure 5.11.2.5.1 Location Page, references* [*2*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.1rf9gpq)*,* [*8*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.jzpmwk)*,* [*9*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.33zd5kd)*, and* [*10*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#bookmark=id.vpp6j315edbo) *from* [*5.11.1.1*](https://docs.google.com/document/d/1nXmrUuXUBkf54qWdp3ubsKHYQHdU2dCR/edit#heading=h.3c9z6hx)

*The green circles reference the items below*

1. From the Location Page ([8](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1csj400)), if the user taps a picture of that location, the app shall transition to the respective lightbox ([9](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.3ws6mnt)) containing the image.
2. From the Location Page ([8](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.1csj400)), if the user presses the home button ([10](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.3t3zi4wtlf68)), the app shall return to the Home Page ([2](https://docs.google.com/document/d/1t9csnwGjo55t2-Dy9yTQ3n13y688w5Px/edit#bookmark=id.2k82xt6)).

# 

# 

# 5.12 Algorithm

According to IEEE 1016-2009 section 5.12 page 21, the algorithm viewpoint is “the detailed design description of operations (such as methods and functions), the internal details and logic of each design entity.” Trailru uses Javascript, NodeJS and SQLite to handle functionality, including accessing and storing relevant data inside a MySQL database. Pseudocode in this section will show how Trailru will handle functionality and data.

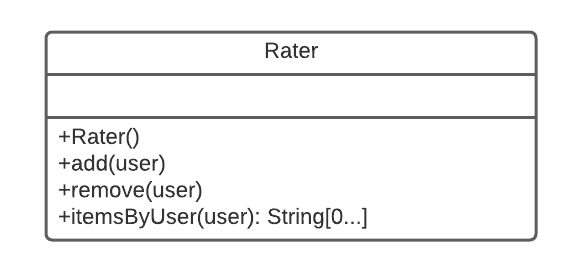
### 5.12.1 Hike Recommendation Algorithm (SRS [1.2.2](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.9hoof3hocj0g), [3.2.7](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.va8hejdnf2z2), [3.4.6](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.amsuym6zw4yz))

The Hike Recommendation Algorithm is the system of classes and functions that allows Trailru’s users to receive personalized recommendations for hiking trails. In order to accomplish this, the Hike Recommendation Algorithm has several pieces: the Rater class which records and stores each user’s actions, the Similar class that processes each user’s actions to find other users with similar actions, and the Suggestion class that takes the similar users and supplies recommendations based on the collected data. All data generated by the Hike Recommendation Algorithm will be stored in the database attached to the [user\_id](#_702f6qrut9q4) of each user. See Figure 5.12.1 for the logical flow of the function, and how each piece fits together.

### 

*Figure 5.12.1: Flowchart depicting the logical flow of the Trail Recommendation Algorithm*

#### 5.12.1.1 Rater Class

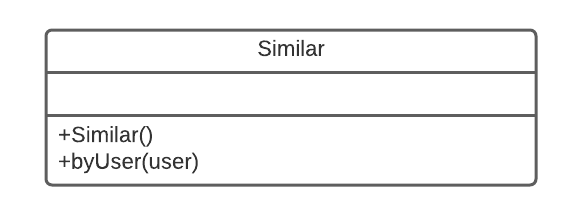


*Figure 5.12.1.1: Diagram of the Rater class*

The Rater class’s sole purpose is to record a user’s actions as they occur. These actions include liking and disliking trails, as well as the user physically hiking a trail. Each type of action will be stored separately, and so different instances of the class are needed. This means that there will be 3 instances of the rater class: PreferenceTrailRater, HikedTrailRater, and EquipmentTrailRater (SRS [3.2.7.1](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.oe1py91ziutt), [3.2.7.2](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.h47g4i6w161v), and [3.2.7.3](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit#bookmark=id.ukapphpfg2a) respectively).

The functions of the rater class are simple. When the user performs an action, the rater class calls the add() function to add it to the database. If the user undoes said action within a set amount of time (a default of 10 seconds), the remove() function will then remove that action from the database. The itemsByUser() function accepts a user as an argument, and will return the array of actions performed by that user.

#### 5.12.1.2 Similar Class



*Figure 5.12.1.2: Diagram of the Similar class*

The Similar class takes the actions that have been recorded by the rater instances and generates [similarity indexes](#1t2qco2s1try) from that data. First, the byUser() function is called, and a user is passed in. The Similar class then scans the database and compares the user’s actions to others, and generates a list of similar users. Users with a similarity score of .85 or higher are stored as similar users without question. If no users with a score of .85 or higher are found, then users with a score of .60 may be used instead. The selected users are then stored in the similarity array by their user id and their similarity score.

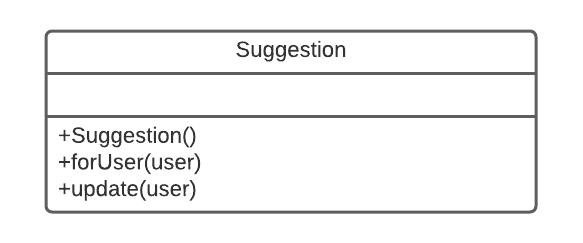
The Similar class does not use any of this data, it simply generates and stores it for other classes to use.

##### 5.12.1.2.1 byUser Function

|  |
| --- |
| function byUser(user)  {  Create an array that stores pairs, of type User and type Float  Get the user's liked trails, hiked trails, and equipment    Find others who have liked the same trails, hiked the same trails, and own the same equipment    Calculate the similarity index of the others found in the previous step    Store anyone with a similarity index >= .85 in the array    If the array is still empty  Store users with a similarity index >= .60 in the array    endif  Store the array in the database in a hashmap, with the key being the user\_id    end  } |

#### 

#### 5.12.1.3 Suggestion Class



*Figure 5.12.1.3: Diagram of the Suggestion class*

The Suggestion class takes the data aggregated and processed by the rater and similar classes, and generates suggestions from it. If the user has no previous recommendations, the forUser() function is called, which generates those from scratch. If the user already has recommendations that are stored in the database, the update() function checks if the user has any new similar users, and updates the suggestions accordingly.

##### 5.12.1.3.1 forUser Function

|  |
| --- |
| function forUser(user)  {  Create an array to store recommendations  Check the hashmap in the database to see if the user has any similar users  If no similar users are found  Add the 3 most popular trails to the array  endif  Else    Pull all of the the similar user's liked trails from the database  Count the number of similar users that liked each trail  If at least one trail was liked more than once  Add up to 3 to the recommendation array, starting with the highest liked trail  endif  If no trails were liked more than once  Randomly select three to recommend from among the similar user’s liked trails  endif  Store recommendation array in database, attached to user's user\_id  endelse  Display the trails found in the recommendation array  end  } |

### [5.12.2](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit" \l "bookmark=id.q7bnhq21pq39) Keyword Search

Relevant keywords are found in the trails table. The following pseudocode will serve as a guide on how to query these keywords. See section [5.6.2.2](#k5ua8f3x3rxb) for table entity details.

QUERY keyword FROM all columns WITHIN TABLE trails

DISPLAY results TO presentation view.

### [5.12.3](https://docs.google.com/document/d/1bT6ELQNTdg1O-alrJfPAo3woqcuZAi8VbPy_bWSeuNo/edit" \l "bookmark=id.q7bnhq21pq39) Trail Search

Trail searching will query exact or near exact names. The following pseudocode will serve as a guide on how to query these keywords. See section [5.6.2.2](#k5ua8f3x3rxb) for table entity details.

QUERY keyword FROM trail\_name WITHIN TABLE trails

DISPLAY results TO presentation view.

# 5.13 Resources

The resource viewpoint is a view of the performance of the available resources in the Trailru application. It lists the available resources in the diagrams and the performance of how the resources interact with each other. Each diagram is another aspect of the performance of the application and is described in detail below. The resource viewpoint is described in the IEEE Std 1016-2009 on page 22, section 5.13.

### 5.13.1 Search results in 5 seconds or less

Related SDD Sections: [5.2.1](#51ewvu4x0c8z), [5.8.1.1](#j07qy2jvvzv1), [5.11.1](#6gc6oodxlzzw)

Design Concerns from SRS: [1.2.1](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79)

Results searched for by the user shall be shown in 5 seconds or less to the user. This constraint shall be consistent 95% of the time during testing. The process is complete when the user can scroll through the results of the search. This applies to any hike or word that the user searches for.

|  |
| --- |
|  |
| *Figure 5.13.1.1 Search Results Diagram* |

### 5.13.2 Traffic level view in 3 seconds or less

Related SDD Sections: [5.2.3](#1ri48hfthxmc)

Design Concerns from SRS: [1.2.3](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79)

The user will be able to view the traffic level of the hike from the hiking selection page within 3 seconds. During the testing phase this process shall be completed in 3 seconds or less 95% of the time. The process is complete when the user can see the traffic level of the hike.

### 

|  |
| --- |
|  |
| *Figure 5.13.2.1 Traffic Level View Diagram* |

### 5.13.3 Hike distance and difficulty view in 3 seconds or less

Related SDD Sections: [5.2.5](#izi97lri6o7r)

Design Concerns from SRS: [1.2.5](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79)

The user shall be able to see the distance and the difficulty of the hike in 3 seconds or less. The process shall be complete in 3 seconds or less 95% of the time during the testing phase. The process is complete when the user can see the distance and difficulty of the hike.

|  |
| --- |
|  |
| *Figure 5.13.3.1 Hike Distance and Difficulty Diagram* |

### 5.13.4 Maps shall be downloaded in 30 seconds or less

Related SDD Sections: [5.2.8](#6ktzsx7bora6), [5.4.5](#u7mi2g2mmmsk), [5.4.6](#j96zu0jiue4a), [5.9.5](#wjwpi4ewfi7n), [5.9.6](#acxnk9f528vm)

Design Concerns from SRS: [1.2.8](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79)

The user can download maps through Trailru for hiking in areas where internet connection is weak within 30 seconds. During the testing phase of this process shall be completed in 30 seconds or less 90% of the time. The process is complete when the map is downloaded and visible to the user on their phone.

|  |
| --- |
|  |
| *Figure 5.13.4.1 Map Downloads Diagram* |

### 5.13.5 Hiking equipment recommendations in 5 seconds or less

Related SDD Sections: [5.2.11](#yk16lbj265t1)

Design Concerns from SRS: [1.2.11](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79)

The user shall be able to view the suggested equipment for any hike in 5 seconds or less. This shall take less than 5 seconds 95% of the time while testing. The process is complete when the user can scroll through the list of suggested equipment and be interacted with by the user.

|  |
| --- |
|  |
| *Figure 5.13.5.1 Hiking Equipment Recommendations Diagram* |

### 5.13.6 Phone camera access in 3 seconds or less

Related SDD Sections: [5.2.14](#xqphz3r58uat), [5.8.3.2](#9cpd4ksu8sva)

Design Concerns from SRS: [1.2.14](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79)

The user shall be able to access the camera of their phone through a button on Trailru within 3 seconds. During the testing phase this process shall be completed in 3 seconds or less 95% of the time. The process is complete when the user can take pictures with their phone’s camera app.

|  |
| --- |
|  |
| *Figure 5.13.6.1 Phone Camera Access Diagram* |

### 5.13.7 Connection to API’s is done in 3 second or less

Related SDD Sections: [5.3.4.1](#kzfqyg5syh2p), [5.3.4](#ijuv2xpw5pe6)

Design Concerns from SRS: [1.3.1.4](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79), [3.1.2](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/view#heading=h.6o0fo4387r79)

The app shall use different API’s such as, weather, maps, advertisements, and safety to provide information to the user. The app shall have access to Maps [SDK](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/edit#bookmark=id.r7bnxd61taeq) for Android (v3.10.0) and the Maps [SDK](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/edit#bookmark=id.r7bnxd61taeq) for [iOS](https://docs.google.com/document/d/1hSwX4F2dQ-_aW1tYD_gHZ_BDYtgEM0MuNzK4ujfMcGQ/edit#bookmark=id.e10qvl9m9doz) (v3.10.0). These connections shall take place within 1 second. During the testing phase this connection shall be complete in 1 second or less 90% of the time. The connection is complete when the application can use the API to render the information onto the screen.

|  |
| --- |
|  |
| *Figure 5.13.7.1 API Connection Diagram* |