SwimFit

SwimFit

Trainingsplanerstellung

Trainingsauswertung durch Pulsmessung

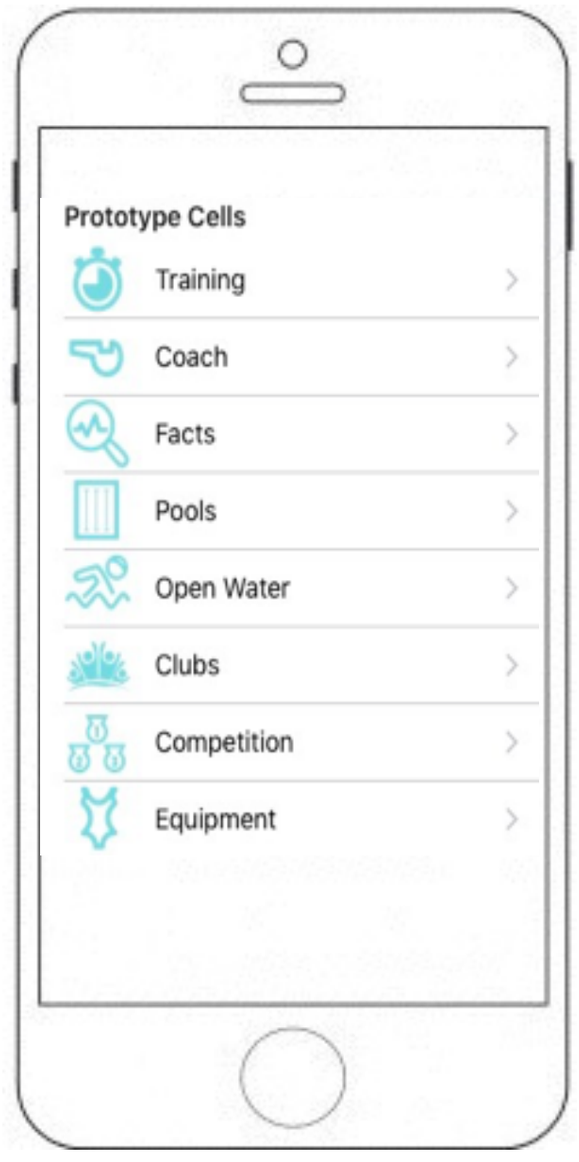
Schwimmmöglichkeiten in der Nähe anzeigen

Wettbewerbsvorbereitung und Informationen

Trainingsvideos

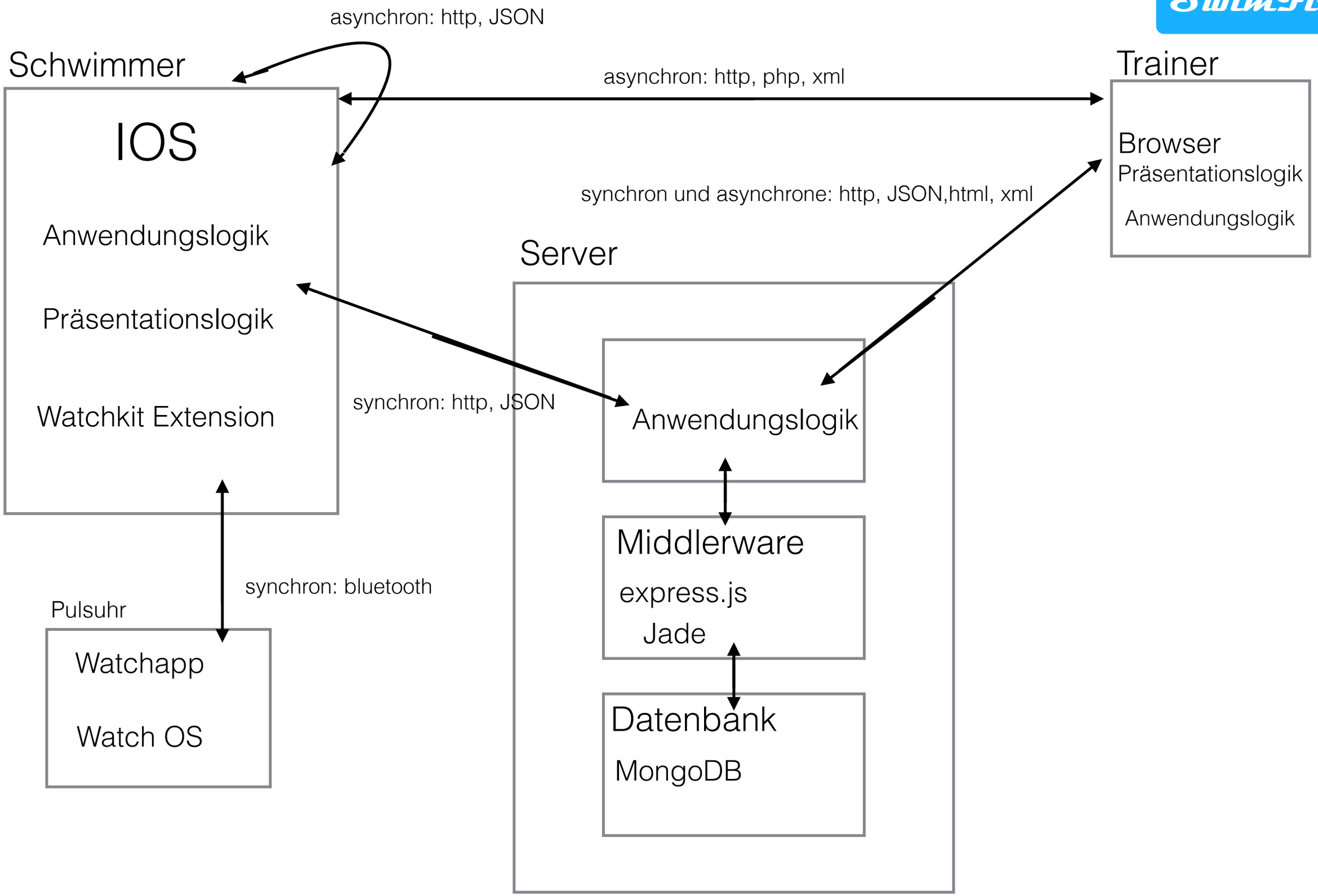
Austausch mit anderen Schwimmern und Trainern

SwimFit


Prototype Cells
Training
Coach
Facts
Pools
Open Water
Clubs
Competition
Equipment


10:25
1.59MI
152BPM
Heart Rate

Architektur



Ressourcen Beispiel

Ressourcen	URI	Methode
Liste aller Schwimmer	/schwimmer	GET,POST
einzelner Schwimmer	/schwimmer/{id}	GET,PUT,POST,DELET
Liste aller Schwimmer eines Vereins	/schwimmer?verein=unisport	GET
einzelnes Trainingsmodul	/Modul/{id}	GET,PUT,POST,DELET
einzelner Trainingsplan	/trainingsplan/{id}	GET
Schwimmvereine	/clubs	GET,PUT,POST,DELET

Datenstruktur Trainingsmodul

{
modulname: “E1“,
ModulGruppe:“A“
Strecke:150,
Lage: “50 B-Arme / K-Beine+ 50 R- Altdeutsch+ 50 ganze“,
Zeit:3:30,
Puls: 140
}

Anzahl	Strecke (m)	Lage	Puls	Summe	Gesamt	Zeit
Einschwimmen						
1	300	bel	110	300	300	06:00
3	150	50 B-Arme / K-Beine+ 50 R- Altdeutsch+ 50 ganze Lage	120	450	750	03:30
4	25	Brust Technik	140	100	850	00:40
Hauptteil						
18	50	6xMini-Lagen, 1 min Pause, 6xHL, 1min Pause, 6x NL	140	900	1750	01:00
1	400	HL - Technik	135	400	2150	10:00
5	100	HL 25schnell+75locker, 50schnell+50locker, 75schnell+25locker, 100locker, 100schnell	140	500	2650	02:00
4	50	50 Brust gleiten, so wenig Armzüge wie möglich	130	200	2850	00:00
1	400	bel. Beine	140	400	3250	10:00
1	400	bel. Arme mit Paddles und Pullboy	140	400	3650	07:00
Ausschwimmen						
1	200	locker	110	200	3850	00:00
				0	3850	00:00

Trainingsmodul abrufen


```
- (void)import
{
    NSURL* url = [NSURL URLWithString:[kBaseURL stringByAppendingPathComponent:kModul];

    NSMutableURLRequest* request = [NSMutableURLRequest requestWithURL:url];
    request.HTTPMethod = @"GET";
    [request addValue:@"application/json" forHTTPHeaderField:@"Accept"];

    NSURLSessionConfiguration* config = [NSURLSessionConfiguration defaultSessionConfiguration];
    NSURLSession* session = [NSURLSession sessionWithConfiguration:config];

    NSURLSessionDataTask* dataTask = [session dataTaskWithRequest:request completionHandler:^(NSData *data, NSURLResponse *response, NSError *error) {
        if (error == nil) {
            NSArray* responseArray = [NSJSONSerialization JSONObjectWithData:data options:0 error:NULL];
            [self parseAndAddModul:responseArray toArray:self.objects];
        }
    }];

    [dataTask resume];
}

- (void)parseAndAddModul:(NSArray*)modul toArray:(NSMutableArray*)destinationArray
{
    for (NSDictionary* item in locations) {
        Modul* modul = [[Modul alloc] initWithDictionary:item];
        [destinationArray addObject:modul];
    }

    if (self.delegate) {
        [self.delegate modelUpdated];
    }
}
```

Objekt Trainingsmodul erstellen

```
- (instancetype) initWithDictionary:(NSDictionary*)dictionary
{
    self = [super init];
    if (self) {
        self.modulname = dictionary[@"modulname"];
        self.modulgruppe = dictionary[@"gruppe"];
        self.strecke = dictionary[@"strecke"];
        self.lage = dictionary[@"lage"];
        self.zeit = dictionary[@„zeit“];
        self.puls = dictionary[@"puls"];
        _categories = [NSMutableArray arrayWithArray:dictionary[@"categories"]];
    }
    return self;
}
```

Schwimmer speichern


```
- (void) persist:(Schwimmer*)schwimmer
{
    if (!schwimmer || schwimmer.name == nil || schwimmer.name.length == 0) {
        return;
    }


    NSString* schwimmer = [kBaseURL stringByAppendingPathComponent:kSchwimmer];

    BOOL isExistingSchwimmer = schwimmer._id != nil;
    NSURL* url = isExistingSchwimmer ? [NSURL URLWithString:[schwimmer stringByAppendingPathComponent:schwimmer._id]] :
    [NSURL URLWithString:schwimmer];

    NSMutableURLRequest* request = [NSMutableURLRequest requestWithURL:url];
    request.HTTPMethod = isExistingSchwimmer ? @"PUT" : @"POST";

    NSData* data = [NSJSONSerialization dataWithJSONObject:[schwimmer toDictionary] options:0 error:NULL];
    request.HTTPBody = data;

    [request addValue:@"application/json" forHTTPHeaderField:@"Content-Type"];

    NSURLSessionConfiguration* config = [NSURLSessionConfiguration defaultSessionConfiguration];
    NSURLSession* session = [NSURLSession sessionWithConfiguration:config];

    NSURLSessionDataTask* dataTask = [session dataTaskWithRequest:request completionHandler:^(NSData *data, NSURLResponse *response, NSError *error) {
        if (!error) {
            NSArray* responseArray = @[[NSJSONSerialization JSONObjectWithData:data options:0 error:NULL]];
            [self parseAndAddSchwimmer:responseArray toArray:self.objects];
        }
    }];
    [dataTask resume];
}
```

Schwimmerobjekt in JSON umwandeln


```
#define safeSet(d,k,v) if (v) d[k] = v;
- (NSDictionary*) toDictionary
{
    NSMutableDictionary* jsonable = [NSMutableDictionary dictionary];
    safeSet(jsonable, @"name", self.name);
    safeSet(jsonable, @"geburtstag", self.geburtstag);
    safeSet(jsonable, @"geschlecht", self.geschlecht);
    safeSet(jsonable, @"kenntnisse", self.kenntnisse);
    safeSet(jsonable, @"verein", self.verein);
    safeSet(jsonable, @"ruhepuls", self.ruhepuls);
    return jsonable;
}
```