\*\*No changes were necessary from Week 2, so this document is being submitted as-is.

My Vacation & Timeoff Tracker application will use RESTful services and HTTP protocol to communicate with my relational database schema, and it will use JSON format to do so.

Below are the various RESTful endpoints along with the proposed JSON code:

**student**

1. Upon initial login and registration to the application:

POST /student

{  
 “studentID”: 0,  
 “firstName”: st\_fname,  
 “lastName”: st\_lname,  
 “startDate”: startdate,  
 “email”: st\_email,  
 “vacationHrs”: vac\_hrs,  
 “sickHrs”: sick\_hrs,  
 “personalHrs”: per\_hrs  
}

Response Codes  
200 – success  
400 – invalid studentID supplied  
404 – student not found  
405 – student validation exception (\*\*primary key constraint)

1. The student record will then be retrieved and used throughout the app:

GET /student/{studentID}

{  
 “studentID”: studentID,  
 “firstName”: st\_fname,  
 “lastName”: st\_lname,  
 “startDate”: startdate,  
 “email”: st\_email,  
 “vacationHrs”: vac\_hrs,  
 “sickHrs”: sick\_hrs,  
 “personalHrs”: per\_hrs  
}

Response Codes  
200 – success  
400 – invalid studentID supplied  
404 – student not found

**timeoff**

1. Retrieve timeoff requests and present them to the student in chart, graph, and report forms.

GET /timeoff/{studentID}

{  
 “timeoffID”: timeoffID  
 “studentID”: studentID,  
 “timeoffDate”: toff\_date,  
 “timeoffCd”: toff\_cd,  
 “timeoffHrs”: toff\_hrs,  
 “comments”: toff\_cmnt,  
 “approvalReq”: req\_appr,  
 “managerID”: managerID,  
 “managerAppr”: TRUE/FALSE,  
 “managerApprDate”: CURRENT\_DATE  
}

Response Codes  
200 – success  
400 – invalid timeoffID supplied  
404 – timeoffID request not found

1. Students will submit their new vacation & timeoff requests:

POST /timeoff

{  
 “timeoffID”: 0,  
 “studentID”: studentID,  
 “timeoffDate”: toff\_date,  
 “timeoffCd”: toff\_cd,  
 “timeoffHrs”: toff\_hrs,  
 “comments”: toff\_cmnt,  
 “approvalReq”: req\_appr,  
 “managerAppr”: TRUE/FALSE,  
 “managerApprDate”: CURRENT\_DATE  
}

Response Codes  
200 – success  
400 – invalid timeoffID supplied  
404 – timeoffID request not found  
405 – timeoffID validation exception

1. The student can also update current and future dated timeoff requests:

PUT /timeoff/{timeoffID}

{  
 “timeoffID”: timeoffID,  
 “timeoffDate”: toff\_date,  
 “timeoffCd”: toff\_cd,  
 “timeoffHrs”: toff\_hrs,  
 “comments”: toff\_cmnt,  
 “approvalReq”: req\_appr,  
 “managerAppr”: TRUE/FALSE,  
 “managerApprDate”: CURRENT\_DATE  
}

Response Codes  
200 – success  
404 – timeoffID not found  
405 – timeoffID validation exception

1. The student can also delete current and future dated timeoff requests:

DELETE /timeoff/{timeoffID}

{  
 “timeoffID”: timeoffID   
}

Response Codes  
200 – success  
400 – invalid timeoffID supplied  
404 – timeoffID not found

**timeoffCd**

There are no RESTful endpoints to maintain the timeoffCd domain table via the Vacation & Timeoff Tracker application. Data will be manually inserted into this table via PostgreSQL.

**approvals**

1. As part of the Admin pages, managers e.g. professors will assign themselves to students by referencing their studentIDs:

POST /approvals

{  
 “managerID”: managerID,  
 “studentID”: studentID,  
 “activeInd”: “**TRUE**”  
}

Response Codes  
200 – success  
400 – invalid info supplied  
404 – info request not found

1. Manager e.g. Professor can also mark relationships inactive when courses end or staffing changes occur:

PUT /approvals/{managerID, studentID}

{  
 “activeInd”: “**FALSE**"  
}

Response Codes  
200 – success  
400 – invalid timeoffID supplied  
404 – timeoffID request not found  
405 – timeoffID validation exception

**manager**

1. As part of the Admin pages, managers e.g. professors will be able to maintain their own account profiles:

POST /manager

{  
 “managerID”: managerID,  
 “firstName”: mgr\_fname,  
 “lastName”: mgr\_lname,  
 “email”: mgr\_email  
}

Response Codes  
200 – success  
400 – invalid managerID supplied  
404 – managerID request not found

1. Manager e.g. Professor can also update their account profiles:

PUT /manager/{managerID}

{  
 “managerID”: managerID,  
 “firstName”: mgr\_fname,  
 “lastName”: mgr\_lname,  
 “email”: mgr\_email  
}

Response Codes  
200 – success  
400 – invalid managerID supplied  
404 – managerID request not found  
405 – managerID validation exception

1. Manager e.g. Professor can also delete their account profiles, thus removing them from the manager and approvals tables:

DELETE /manager/{managerID}

DELETE /approvals/{managerID}

{  
 “managerID”: managerID  
}

Response Codes  
200 – success  
400 – invalid managerID supplied  
404 – managerID not found

**preferences**

1. When the student table entry is created, the preferences table will also be inserted into for vacation recommendations and marketing:

POST /preferences

{  
 “studentID”: 0,  
 “optin”: opt\_in,  
 “domTravel”: dom\_travel,  
 “intTravel”: int\_travel,  
 “resorts”: resorts,  
 “roughingIt”: rough\_it,  
 “fishing”: fish,  
 “hunting”: hunt,  
 “foodie”: food  
}

Response Codes  
200 – success  
400 – invalid studentID supplied  
404 – studentID request not found

1. Students can update their preferences to 1) opt in/opt out without changing their interest elections or 2) update their interest elections while remaining opted in:

PUT /preferences/{studentID}

{  
 “studentID”: studentID,  
 “optin”: opt\_in,  
 “domTravel”: dom\_travel,  
 “intTravel”: int\_travel,  
 “resorts”: resorts,  
 “roughingIt”: rough\_it,  
 “fishing”: fish,  
 “hunting”: hunt,  
 “foodie”: food  
}

Response Codes  
200 – success  
400 – invalid studentID supplied  
404 – studentID request not found  
405 – studentID validation exception

1. Deleting student preferences is not an option. Students must opt out in order to stop receiving marketing messages.

**admin**

There are no RESTful endpoints to maintain the admin table via the Vacation & Timeoff Tracker application. Data will be manually inserted into this table via PostgreSQL.

**studentTrip**

1. There will be a portal for students to add their completed trips to the recommendations and marketing business logic of the application:

POST /studentTrip

{  
 “tripID”: 0,  
 “studentID”: studentID,  
 “timeoffDate”: toff\_date,  
 “rating”: rating,  
 “domTravel”: dom\_travel,  
 “intTravel”: int\_travel,  
 “resorts”: resorts,  
 “roughingIt”: rough\_it,  
 “fishing”: fish,  
 “hunting”: hunt,  
 “foodie”: food  
}

Response Codes  
200 – success  
400 – invalid tripID supplied  
404 – tripID request not found

1. Students can update their completed trips with comments and blogging:

PUT /studentTrip/{tripID}

{  
 “tripID”: 0,  
 “timeoffDate”: toff\_date,  
 “rating”: rating,  
 “domTravel”: dom\_travel,  
 “intTravel”: int\_travel,  
 “resorts”: resorts,  
 “roughingIt”: rough\_it,  
 “fishing”: fish,  
 “hunting”: hunt,  
 “foodie”: food  
}

Response Codes  
200 – success  
400 – invalid tripID supplied  
404 – tripID request not found  
405 – tripID validation exception

1. Students can also delete their completed trips if they so choose, which will remove them from recommendations and marketing business logic:

DELETE /studentTrip/{tripID}

{  
 “tripID”: tripID  
}

Response Codes  
200 – success  
400 – invalid tripID supplied  
404 – tripID not found

**tripPics**

1. Completed trip pictures will be uploaded and stored in the tripPics table:

POST /tripPics

{  
 “tripID”: tripID,  
 “picID”: 0,  
 “tripPic”: trip\_pic  
}

Response Codes  
200 – success  
400 – invalid tripID supplied  
404 – tripID request not found

1. Students can update their completed trips with new pictures:

PUT /tripPics/{tripID}

{  
 “tripID”: tripID,  
 “picID”: 0,  
 “tripPic”: trip\_pic  
}

Response Codes  
200 – success  
400 – invalid tripID supplied  
404 – tripID request not found  
405 – tripID validation exception

1. Students can also delete specific pictures they’ve uploaded or the entire trip:

DELETE /tripPics/{picID}

{  
 “picID”: picID  
}

Response Codes  
200 – success  
400 – invalid picID supplied  
404 – picID not found

DELETE /tripPics/{tripID}

{  
 “tripID”: tripID  
}

Response Codes  
200 – success  
400 – invalid tripID supplied  
404 – tripID not found